



**PEDRO SAMUEL
NUNES DA SILVA**

**Caracterização experimental de comunicações
Powerline em automóveis**



**PEDRO SAMUEL
NUNES DA SILVA**

**Caracterização experimental de comunicações
Powerline em automóveis**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Doutor Luís Miguel Pinho de Almeida, Professor Associado do Departamento de Engenharia Electrotécnica e de Computadores da Universidade do Porto e do Doutor Paulo Bacelar Reis Pedreiras, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Aos meus pais e à Carina...

o júri

presidente

Prof. Doutor José Alberto Gouveia Fonseca

Professor Associado do Dep. de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Prof. Doutor Paulo José Lopes Machado Portugal

Professor Auxiliar do Dep. de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Prof. Doutor Luis Miguel Pinho de Almeida

Professor Associado do Dep. de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Prof. Doutor Paulo Bacelar Reis Pedreiras

Professor Auxiliar do Dep. de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

agradecimentos

A realização deste trabalho envolveu várias pessoas que, directa ou indirectamente contribuíram para o seu sucesso. A todos eles o meu sincero agradecimento.

Aos meus orientadores, Prof. Doutor Luís Almeida e Prof. Doutor Paulo Pedreiras, pela confiança depositada e pelas oportunidades oferecidas. As palavras de incentivo, os erros apontados, as sugestões dadas que em tudo ajudaram a levar cada dia com empenho e vontade de fazer mais e melhor.

Aos meus pais por lutarem todos os dias para me abrirem todas as portas, para me darem o mundo, por confiarem em mim da forma que confiam, por estarem sempre tão longe mas ao mesmo tempo tão perto, por me apoiarem em todas as decisões mais ou menos difíceis... Por tudo! Obrigado por serem quem são e por todos os esforços que sei que fizeram para que possa ter chegado até aqui.

À minha irmã, pelo que sempre me ensinou, pelas oportunidades que me deu e as coisas que me levou a conhecer, por ser uma grande amiga e pela cabeça partida...

Aos meus amigos que me proporcionaram grandes momentos, que de alguma forma me fizeram crescer! Aos amigos das jantaradas, aos que só dizem asneiras (e são muitos!), aos que foram para longe e aos que estão perto todos os dias...

À Carina, por me ouvir quando preciso, por me dar na cabeça quando mereço, por estar sempre ao meu lado. Pelo apoio nas decisões e problemas que teimo em tornar dez vezes mais difíceis do que realmente são. Pelo seu sorriso, pelas suas tolices de todos os dias e pela sua alegria, por me fazer acordar cada manhã com um sorriso e, principalmente, por ser como é.

Finally, a big thanks to my international tutors (Professor Francesco Benzi and Professor Tullio Facchinetti) and friends (Daniele and Alberto) and to Yaniv Serif from Yamar Electronics.

Obrigado a todos!

palavras-chave

Comunicação Powerline, redes de dados para automóveis, comunicação de tempo-real, sistemas de tempo-real.

resumo

O crescimento em quantidade e complexidade dos sistemas electrónicos nos veículos conduziu a uma utilização em larga escala de arquitecturas distribuídas recorrendo a redes específicas como CAN, LIN, TTP/C e mais recentemente FlexRay. Nos últimos tempos uma tecnologia que tem sofrido uma grande evolução é a tecnologia Powerline, em que a comunicação se faz sobre os fios de alimentação. Aplicando esta tecnologia a redes de dados para automóveis será possível reduzir significativamente a cablagem existente nos automóveis.

Os testes levados a cabo neste trabalho experimental tiveram como propósito principal averiguar a viabilidade desta tecnologia como substituta da tão usada rede CAN, focando no comportamento temporal. Foram efectuadas medidas temporais sobre o barramento CAN e sobre o barramento Powerline.

As medidas no barramento Powerline foram efectuadas usando uma das soluções disponibilizadas pela empresa Israelita Yamar Electronics®, o modem transdutor DCB500. Com base nesta solução construiu-se um canal de comunicação Powerline sobre o qual foram efectuadas medidas de tempos de transmissão de pacotes de dados e efectuados alguns testes de arbitragem no acesso ao barramento, com e sem prioridades.

Os resultados exibiram algumas características da tecnologia utilizada, nomeadamente o overhead de comunicação, que apenas permite chegar ao throughput nominal com transferências relativamente largas, bem como revelaram a necessidade de aprofundar os testes de arbitragem no acesso ao barramento com prioridades, pois deixaram no ar várias dúvidas quanto à respectiva eficácia, aspecto considerado fundamental para uma solução comparável com CAN.

keywords

Powerline communication, automotive networks, real-time communication, real-time systems

abstract

The growth in volume and complexity of electronic systems in vehicles has led to a large-scale use of distributed architectures using specific networks such as CAN, LIN, TTP/C and more recently FlexRay. Lately a technology that has suffered major improvements is the Powerline technology. With this technology applied to automotive networks the wiring existing in cars could be significantly reduced.

Tests carried out in this experimental work aimed mainly at determining the feasibility of this technology as a replacement of an automotive CAN network, focusing on its temporal behavior. Temporal measures were made on the CAN and Powerline buses.

Powerline measures on the bus were made using one of the solutions offered by Yamar Electronics®, namely the modem-transducer DCB500. A Powerline communication channel was built using this solution in which time measurements of transmitting packets of data were carried out together with bus access arbitration tests, with and without priorities.

The results exhibited some characteristics of this technology, namely the communication overhead that requires reasonably long transfers to achieve the nominal throughputs, but also revealed the need for further tests concerning the bus access arbitration with priorities, an issue that is considered is fundamental for a solution comparable to the CAN bus.

Índice

ÍNDICE	XIII
LISTA DE FIGURAS	XVII
LISTA DE SIGLAS E ACRÓNIMOS	XIX
ÍNDICE DE TABELAS	XXI
1. INTRODUÇÃO	1
1.1. AS COMUNICAÇÕES NO DOMÍNIO AUTOMÓVEL	3
1.2. A ALTERNATIVA POWERLINE	4
1.2.1. Evolução das tecnologias PLC para AC e opções existentes no mercado	5
1.2.1.1. X10	6
1.2.1.2. HomePlug	7
1.2.2. A solução PLC para DC e a tecnologia YAMAR	9
1.3. OBJECTIVO DA TESE	11
1.4. PARCEIROS	12
1.5. ESTRUTURA DA TESE	12
2. REDES UTILIZADAS NO DOMÍNIO AUTOMÓVEL	15
2.1. CAN	17
2.1.1. Mensagens CAN	18
2.1.1.1. Data Frame	18
2.1.1.2. Remote Frame	19
2.1.1.3. Error Frame	20
2.1.1.4. Overload Frame	20
2.1.2. O processo de arbitragem	21
2.2. LIN	21
2.2.1. Mensagens LIN	22
2.2.1.1. Unconditional Frame	23
2.2.1.2. Event-Triggered Frame	23
2.2.1.3. Sporadic Frame	23
2.2.1.4. Diagnostic Frame	23
2.3. FLEXRAY	24
2.3.1. A ECU FlexRay	25
2.3.2. A Frame FlexRay	25
2.3.2.1. Header	26
2.3.2.2. Payload	26
2.3.2.3. Trailer	26

2.4. A TECNOLOGIA DC-POWERLINE YAMAR	26
2.4.1. SIG40	27
2.4.2. DCAN250	28
2.4.3. DCB500	29
2.4.3.1. Interfaces	30
2.4.3.2. Correção de erros	30
2.4.3.3. Detecção de colisões	30
2.4.3.4. DCB500 Evaluation Board	31
3. CARACTERIZAÇÃO TEMPORAL DO CANAL DE COMUNICAÇÃO CAN	33
3.1. UM EMISSOR, UM RECEPTOR	37
3.1.1. ID701, T=5ms, Data Field constante	37
3.1.2. ID701, T=8ms, Data Field constant	38
3.1.3. ID701, T=5ms, Data Field variável	38
3.1.4. ID701, T=8ms, Data Field variável	39
3.2. DOIS EMISSORES, UM RECEPTOR	40
3.2.1. ID701 e ID702, T=5ms, Data Field constante	41
3.2.2. ID701, T=8ms e ID702, T=5ms, Data Field constante	42
3.2.3. ID701 e ID702, T=5ms, Data Field variável	43
3.2.4. ID701, T=8ms e ID702, T=5ms, Data Field variável	44
3.3. TRÊS EMISSORES, UM RECEPTOR	45
3.3.1. ID701, ID702 e ID 703, T=5ms, Data Field constante	46
3.3.2. ID701, T=8ms, ID702 e ID 703, T=5ms, Data Field constante	47
3.3.3. ID701, ID702 e ID 703, T=5ms, Data Field variável	48
3.3.4. ID701, T=8ms, ID702 e ID 703, T=5ms, Data Field variável	50
4. CARACTERIZAÇÃO TEMPORAL DO CANAL DE COMUNICAÇÃO POWERLINE	53
4.1. TEMPO DE TRANSMISSÃO DE UMA MENSAGEM	57
4.2. MEDIÇÃO DO ROUND TRIP DELAY (RTD)	62
4.3. COMPARAÇÃO DAS MEDIDAS EFECTUADAS, COM E SEM ARBITRAGEM	64
4.4. COLISÕES	69
4.4.1. Transmissões em simultâneo	69
4.4.2. Desfasamento maior que o offset de arbitragem	72
4.4.3. Desfasamento menor que o offset de arbitragem	75
5. CONCLUSÕES E TRABALHO FUTURO	77
5.1. CONCLUSÕES E COMENTÁRIOS FINAIS	79
5.2. TRABALHO FUTURO	79

6. REFERÊNCIAS	81
----------------------	----

Lista de Figuras

Fig. 1 – Princípio de modulação do sinal na comunicação Powerline	4
Fig. 2 – Transmissão de frames X10.....	6
Fig. 3 – Transmissão dos símbolos binários no protocolo X10 – Start Of Frame	6
Fig. 4 – Transmissão dos símbolos binários no protocolo X10 – House Code e Unit Code	7
Fig. 5 – Espectro de frequência dos protocolos X10 e HomePlug AV	8
Fig. 6 – Long Frame – HomePlug.....	8
Fig. 7 – Short Frame – HomePlug.....	8
Fig. 8 – Implementação HomePlug AV num ambiente doméstico [11]	9
Fig. 9 – Circuito de interligação CAN, DC-Bus – Modem DC-CAN.....	11
Fig. 10 – Data Frame, identificador de 11 bits.....	19
Fig. 11 – Data Frame, identificador de 29 bits.....	19
Fig. 12 – Remote Frame, do tipo CAN 2.0A	20
Fig. 13 – Error Frame.....	20
Fig. 14 – Processo de arbitragem de acesso ao barramento CAN	21
Fig. 15 – Exemplo de comunicação LIN com dois nós.....	22
Fig. 16 – Frame LIN	23
Fig. 17 – Configuração Dual Channel – FlexRay	24
Fig. 18 – Exemplo de uma Electronic Control Unit FlexRay.....	25
Fig. 19 – Frame FlexRay	25
Fig. 20 – Exemplo de configuração de rede baseada em Evaluation Boards DCB500 [28]	27
Fig. 21 – Exemplo de uma rede baseada no SIG40	28
Fig. 22 – Ilustração da extracção do sinal de dados da linha de alimentação, com o SIG40.....	28
Fig. 23 – Diagrama de blocos de um DCB500.....	29
Fig. 24 – Exemplo de ligação SPI entre o DCB500 e o Host.....	30
Fig. 25 – Fotografia real de uma EVB DCB500	31
Fig. 26 – Diagrama de blocos da Evaluation Board DCB500.....	31
Fig. 27 – Setup experimental das medições sobre o barramento CAN	35
Fig. 28 – Ilustração do <i>jitter</i> possível nas mensagens recebidas	36
Fig. 29 – ID 701 envia a cada 5ms – DATA = 0x81...81	37
Fig. 30 – ID 701 envia a cada 8ms – DATA = 0x81...81	38
Fig. 31 – ID 701 envia a cada 5ms – DATA = DATA + 0x59	39
Fig. 32 – ID 701 envia a cada 8ms – DATA = DATA + 0x59	39
Fig. 33 – Duas mensagens no acesso ao barramento	40
Fig. 34 – ID 701 e ID 702 enviam a cada 5ms – DATA = 0x81...81	41
Fig. 35 – ID 701 envia a cada 8ms e ID 702 envia a cada 5ms – DATA = 0x81...81	42
Fig. 36 – ID 701 e ID 702 enviam a cada 5ms – DATA = DATA + 0x59	43

Fig. 37 – ID 701 envia a cada 8ms e ID 702 envia a cada 5ms – DATA = DATA + 0x59	44
Fig. 38 – Acesso ao barramento – 3 nós transmissores.....	45
Fig. 39 – ID 701, 702 e 703 enviam a cada 5ms, DATA = 0x81...81.....	47
Fig. 40 – ID 701, T=8ms, ID 702 e 703, T=5ms – DATA = 0x81...81	48
Fig. 41 – ID 701, 702 e 703, T=5ms – DATA = DATA + 0x59	49
Fig. 42 – ID 701, T=8ms e 702 e 703, T=5ms – DATA = DATA + 0x59	51
Fig. 43 – Montagem experimental do canal de comunicação Powerline.....	57
Fig. 44 – Algoritmo para medida do tempo de transmissão de uma mensagem	58
Fig. 45 – Ilustração do processo de medida	58
Fig. 46 – Tempo de transmissão – 1 a 31 bytes – Single Message Mode	59
Fig. 47 – Tempo de transmissão – 1 a 32 bytes – Stream Mode.....	59
Fig. 48 – Tempo de transmissão – 64 e 4096 bytes – Stream Mode	60
Fig. 49 – Bit Rate para mensagens entre 1 e 31 bytes – Single Message Mode	61
Fig. 50 – Bit Rate para mensagens entre 1 e 32 bytes – Stream Mode.....	61
Fig. 51 – Bit Rate para mensagens entre 64 e 4096 bytes – Stream Mode	62
Fig. 52 – Processo de medida do RTD	63
Fig. 53 – RTD medido para mensagens entre 1 e 32 bytes – Stream Mode	63
Fig. 54 – RTD medido para mensagens entre 64 e 4096 bytes – Stream Mode.....	64
Fig. 55 – Tempo de transmissão – 1 a 29 bytes – Single Message Mode – 300Kbps	65
Fig. 56 – Tempo de transmissão – 1 a 29 bytes – Single Message Mode – 500Kbps	65
Fig. 57 – Tempo de transmissão – 1 a 32 bytes – Stream Message Mode – 300 Kbps	66
Fig. 58 – Tempo de transmissão - 64 e 4096 bytes – Stream Message Mode – 300Kbps.....	66
Fig. 59 – Tempo de transmissão – 1 a 32 bytes – Stream Message Mode – 500Kbps.....	67
Fig. 60 – Tempo de transmissão – 64 a 4096 bytes – Stream Message Mode – 500Kbps	67
Fig. 61 – Transmissão de 31 bytes. (a) – Arbitration Disable. (b) – Priority Arbitration.....	68
Fig. 62 – Ilustração de uma colisão e sinalização ao Host	69
Fig. 63 – Algoritmo para a medição de mensagens perdidas – Transmissão simultânea	70
Fig. 64 – Duas transmissões simultâneas – Arbitration Disable.....	70
Fig. 65 – Duas transmissões simultâneas – Random Arbitration	71
Fig. 66 – Duas transmissões simultâneas – Node 1 > priority	71
Fig. 67 – Algoritmo desenvolvido – Desfasamento maior que o offset medido	72
Fig. 68 – Desfasamento maior que o offset, 50µs – Arbitration Disable.....	73
Fig. 69 – Desfasamento maior que o offset, 50µs – Random Arbitration.....	73
Fig. 70 – Desfasamento maior que o offset, 50µs – Node 1 > priority	74
Fig. 71 – Desfasamento menor que o offset, 15µs – Arbitration Disable.....	75
Fig. 72 – Desfasamento menor que o offset, 15µs – Random Arbitration	75
Fig. 73 – Desfasamento menor que o offset, 15µs – Node 1 > priority	76

Lista de Siglas e Acrónimos

AC	- Alternating Current
AD	- Arbitration Disable
AE	- Arbitration Enable
ARQ	- Automatic Repeat reQuest
bps	- bits per second ou bits por segundo
CAN	- Controller Area Network
COTS	- Commercial Off The Shelf
CRC	- Cyclic Redundancy Check
CSMA/CD	- Carrier Sense Multiple Access with Collision Detection
DC	- Direct current
ECC	- Error Correction Code
ECU	- Electronic Control Unit
EOF	- End Of Frame
EVB	- EVAlution Board
FIFO	- First In, First Out
FPGA	- Field Programmable Gate Array
HD	- High Definition
IPTV	- Internet Protocol Television
ISO	- International Organization for Standardization
kbps	- Kilo bits per second ou Kilo bits por segundo – 1000 bps
Kbps	- Kilo bits per second ou Kilo bits por segundo – 1024 bps
LAN	- Local Area Network
LIN	- Local Interconnect Network
MAC	- Medium Access Controller
Mbps	- Mega bits per second ou Mega bits por segundo
MOST	- Media Oriented Systems Transport
OFDM	- Orthogonal Frequency Division Multiplexing
OSI	- Open Systems Interconnection
PC	- Personal Computer
PL	- Power Line
PLC	- Power Line Communication
RPM	- Rotações Por Minuto
SOF	- Start Of Frame
SPI	- Serial Peripheral Interface
UART	- Universal Asynchronous Receiver/Transmitter
USB	- Universal Serial Bus
TDMA	- Time division multiple Access
TV	- Television ou Televisão

VLSI	- Very Large Scale Integration
VoIP	- Voice over Internet Protocol
WCRT	- Worst-Case Response Time

Índice de Tabelas

Tabela 1 – Um emissor, um receptor – Data Field constante	40
Tabela 2 – Um emissor, um receptor – Data Field variável	40
Tabela 3 – Dois emissores, um receptor – Data Field constante	44
Tabela 4 – Dois emissores, um receptor – Data Field variável	45
Tabela 5 – Três emissores, um receptor – Data Field constante	51
Tabela 6 – Três emissores, um receptor – Data Field variável	52
Tabela 7 – Comandos de controlo do modem transdutor DCB500	55
Tabela 8 – Descrição das interrupções do modem DCB500	56
Tabela 9 – Offsets determinados – Single e Stream Mode – 300 e 500Kbps	68

1. INTRODUÇÃO

1.1. As comunicações no domínio automóvel

Bancos climatizados, televisões, consolas de jogos, computadores de bordo, direcção assistida e vidros eléctricos são apenas alguns exemplos dos “luxos” que os automóveis modernos nos proporcionam. Mas nem sempre assim foi! Os chamados “clássicos” não presenteavam os seus condutores com todo este conforto e tecnologia a que os fabricantes de automóveis actuais já tanto nos habituaram e sem a qual já não nos imagináramos viver.

No entanto, para que possamos usufruir de todos estes sistemas, que para a maioria das pessoas são completamente transparentes, grandes desafios de engenharia tiveram que ser ultrapassados e grandes passos tiveram que ser dados.

Inicialmente, com a nula ou escassa existência de electrónica nos automóveis, era relativamente simples conectar todos os subsistemas que os constituíam, através de cabos dedicados. Cada subsistema conectava directamente a outros com quem necessitasse de comunicar. Contudo, com o evoluir da tecnologia, novos subsistemas foram sendo introduzidos nos automóveis, a sua complexidade aumentou, aumentando a quantidade de cablagem necessária para conectar todos os subsistemas, tornando-se impossível efectuar ligações ponto a ponto de cada um dos subsistemas. Surgiu assim a necessidade de criar barramentos de comunicação série (*buses*) capazes de garantir a comunicação entre todos os subsistemas reduzindo simultaneamente a quantidade de cablagem necessária.

Inicialmente, os fabricantes de automóveis começaram por desenvolver os seus próprios *buses*, mas a extensa variedade de barramentos e protocolos não permitia a interacção entre os fabricantes e os produtores de subsistemas. Surgiu assim a necessidade de uniformizar os protocolos de comunicação usados no domínio automóvel [1].

Uma das tecnologias que foi criada, no início dos anos 90 e que rapidamente se tornou a mais usual na indústria automóvel, foi o barramento CAN, que é do tipo *event-triggered*, *multi-master*, efectua comunicação *broadcast* e é dotado de um sofisticado mecanismo de detecção de erros e reenvio automático de mensagens em caso de erro e/ou falhas.

Este protocolo, de acordo com o modelo de comunicação ISO/OSI de sete camadas, é bastante simplificado, que implementa apenas as primeiras duas camadas do modelo, a *Data-Link Layer* e a *Physical Layer*, permitindo assim a cada utilizador desenvolver as suas aplicações de alto nível com uma ligação muito próxima às propriedades lógicas e temporais de CAN.

Assim, surgiram protocolos de nível superior que complementavam o protocolo CAN, tais como DeviceNet, SDS (Smart-Distributed System), CAL (CAN Application Layer), CANOpen, CANKingdom, etc.

Em 1999 surgiu o protocolo LIN (Local Interconnect Network). Este protocolo surgiu como uma alternativa económica usada como sub-rede de uma rede CAN. De pequenas dimensões e baixa velocidade, as redes LIN permitem conectar sensores inteligentes e actuadores nos veículos actuais, começando agora a ser usadas também em comunicação Powerline, sobre as linhas de alimentação, usando transdutores DC-LIN [2].

No ano 2000 surgiu o *FlexRay Consortium*¹. Liderada por sete grandes empresas ligadas ao sector automóvel, esta organização é a responsável pelo desenvolvimento do protocolo *FlexRay*. Surgiu com o objectivo de criar um protocolo de comunicação suficientemente poderoso, capaz de albergar todas as necessidades das aplicações de controlo dos automóveis do futuro. Elevado ritmo de transmissão (até 20Mbps), comportamento *time* e *event-triggered*, tolerância a falhas, redundância e determinismo são os seus maiores trunfos.

A sua primeira utilização em grande escala foi no final de 2006, no BMW X5, dotado de um novo e rápido sistema de amortecimento adaptativo. Em 2008, o novo BMW serie 7, foi o primeiro veículo a ser produzido, a nível mundial, exclusivamente com o uso do protocolo *FlexRay* [3].

No campo de redes de comunicação multimédia para automóveis surgiu o protocolo *MOST* (Media Oriented Systems Transport). Concebido em 1997, é um protocolo criado com o intuito de interligar sistemas multimédia em veículos. A comunicação é feita através de fibra óptica, permitindo assim uma taxa de transmissão elevada. Como principais características, este protocolo, apresenta a fácil introdução e remoção de nós, através da propriedade *Plug and Play*² (*PnP*), tem um nó *Master* que é quem comanda e fornece os dados aos nós *Slave*, possui uma largura de banda de até 23Mbps e possui 60 canais, dos quais 15 são configuráveis pelo utilizador [4].

1.2. A alternativa Powerline

A comunicação Powerline é um sistema de transmissão de dados que permite, num mesmo fio condutor, transportar energia e dados, ou seja, a transmissão de dados é efectuada sobre as linhas de alimentação.

Nos sistemas de energia eléctrica a energia é transportada desde a geração sobre linhas de alta tensão, depois distribuída sobre linhas de média tensão e finalmente utilizada pelo consumidor final a baixa tensão. A alternativa de comunicação Powerline pode ser introduzida em qualquer um destes estágios.

O conceito por trás desta tecnologia consiste na injeção de um sinal, modulado e limitado em amplitude, nas linhas de alimentação, tal como ilustra a Fig. 1.

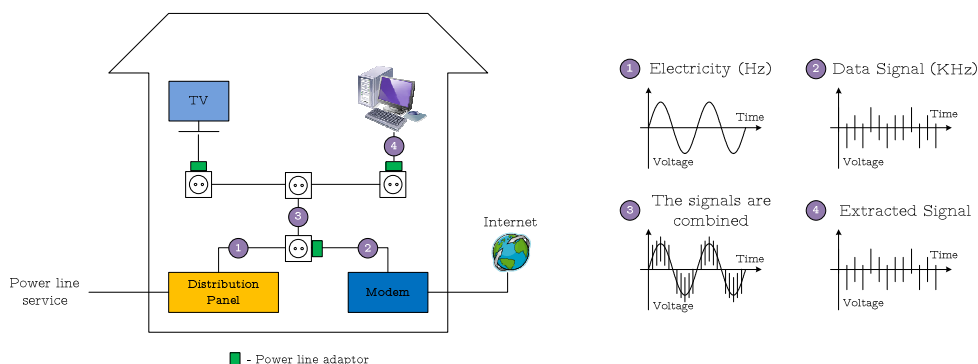


Fig. 1 – Princípio de modulação do sinal na comunicação Powerline

¹ <http://www.flexray.com/>

² **Plug and Play (PnP)**, significa “ligar e usar”. É a propriedade de reconhecimento e configuração automáticos de um dispositivo, permitindo a fácil expansão e eliminando a configuração manual.

Sempre com o mesmo objectivo final, o de permitir a diminuição de cablagem e os seus custos inerentes (material, construção, montagem, etc.), a indústria automóvel tem todo o interesse em adoptar esta nova tecnologia. Isto porque num automóvel moderno existe uma complexa mistura de tecnologias electrónicas e mecânicas, o que leva à existência de grandes quantidades de cablagem. Numerosas unidades de controlo electrónico (ECU) controlam todos os processos cruciais, tais como o sistema de ignição, sistema de travagem, sistema de refrigeração, etc.

Reduzir a quantidade de cablagem terá assim um impacto directo no peso, custos e espaço ocupado. Além disso, a quantidade de cabos é um grande problema na montagem dos automóveis, pois a sua concepção tem que ter em conta os caminhos que seguirão os fios de modo a facilitar a sua instalação. Em alguns campos, como os automóveis de competição, os próprios conectores são muito dispendiosos pois cada cabo transporta dezenas ou mesmo centenas de fios, sendo então muito vantajoso reduzir a quantidade de cabos, que terá assim um impacto directo no custo e fiabilidade dos conectores.

Nas ECU modernas, os conectores e respectivos contactos podem ocupar até 20% do seu tamanho. Assim, reduzir o tamanho dos conectores reduzirá também o tamanho e custo das ECU.

Está assim aberta uma porta ao sucesso de uma tecnologia Powerline para aplicação em automóveis. Visto que, sendo aparelhos electrónicos, as ECUs partilham a linha de alimentação, a comunicação Powerline permitirá retirar os barramentos actualmente existentes, diminuindo assim drasticamente a quantidade de cablagem existente num automóvel.

1.2.1. Evolução das tecnologias PLC para AC e opções existentes no mercado

Inicialmente criada para comunicação sobre linhas AC, a comunicação *Powerline* permite uma vasta gama de taxas de transmissão de dados, dependendo da aplicação à qual é destinada. As suas taxas de transmissão variam desde as centenas de bits por segundo (bps) ou poucas centenas de Kbps em aplicações de controlo e telemetria, sobre linhas de alta tensão, até taxas de transmissão da ordem dos Mbps em aplicações como, por exemplo, Local Area Networks (LANs).

Para baixas taxas de transmissão é possível transportar energia e dados por milhares de quilómetros, conseguindo assim distribuir a comunicação por lugares, primariamente não servidos por comunicações, com um investimento relativamente diminuto, dada a instalação eléctrica já existente. Por outro lado, a altas taxas de transmissão a extensão de rede está bastante mais limitada, por exemplo, a um piso de um prédio de habitação. Contudo, apesar da limitação de extensão, esta tecnologia permite remover a rede dedicada existente ou evitar a instalação de uma nova rede e utilizar apenas as linhas de alimentação sempre presentes num ambiente doméstico ou empresarial [5].

De seguida são apresentadas algumas das soluções de comunicação Powerline existentes, a operar sobre linhas de alimentação AC.

1.2.1.1. X10

Com o intuito de criar um protocolo de comunicação standard para controlo de aplicações domésticas tais como luzes, aparelhos de áudio ou vídeo, electrodomésticos, etc., o protocolo X10 foi um dos primeiros protocolos de comunicação Powerline a ser introduzido no mercado. Lançado em 1978, este protocolo é um protocolo do tipo *broadcast* [6].

Os transmissores e receptores X10 são ligados directamente a tomadas instaladas numa habitação e são conectados aos dispositivos de controlo através de uma ligação RJ11. Cada receptor X10 tem o seu próprio endereço (ID) que é constituído por um código de habitação e um código respeitante à unidade em causa, e responde apenas a comandos que a ele são destinados. Existem 256 códigos de unidade possíveis, podendo várias unidades partilhar o mesmo código.

Uma frame X10 leva 11 ciclos de rede a ser transmitida. Os primeiros dois ciclos representam o início da frame, *Start of Frame* (SOF). Os quatro ciclos seguintes representam o código da habitação e os cinco ciclos seguintes podem transportar tanto o código do receptor a que se destina o comando ou o código do comando a executar pelo receptor. Este bloco – SOF, Código de habitação e Código de receptor ou de comando – deve ser sempre transmitido em duplicado, com três ciclos de rede entre cada grupo de duas frames. Excepções a esta regra são os comandos de aumento e diminuição de brilho, que devem ser transmitidos continuamente, pelo menos duas vezes, sem qualquer ciclo de rede entre eles [7].

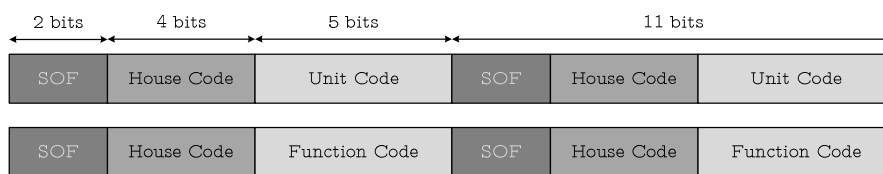


Fig. 2 – Transmissão de frames X10

A transmissão é efectuada sincronamente com o ponto de passagem por zero da linha de alimentação AC. O símbolo binário “1” é transmitido como um pacote de 1ms a 120KHz no ponto de passagem por zero e o “0” binário é transmitido com a ausência desse mesmo pacote, isto é, sem qualquer símbolo.

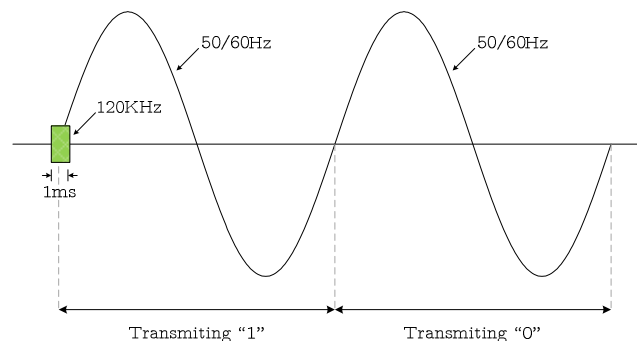


Fig. 3 – Transmissão dos símbolos binários no protocolo X10 – Start Of Frame

Dentro dos blocos de quatro ou cinco ciclos de rede, House Code e Unit Code, cada símbolo deve ser transmitido na sua forma verdadeira e na forma de complemento, na segunda metade do ciclo. Se na primeira metade do ciclo for transmitido um “1”, na segunda metade do ciclo não deve ser transmitido qualquer sinal e vice-versa.

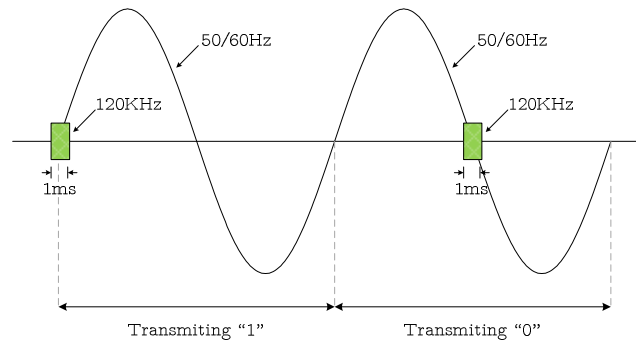


Fig. 4 – Transmissão dos símbolos binários no protocolo X10 – House Code e Unit Code

A simplicidade e disponibilidade tornaram o protocolo X10 o mais conhecido standard no campo da domótica. Usufrui da propriedade *Plug and Play* (PnP) e não requer nenhum conhecimento especial para configurar e manusear a rede doméstica.

Por outro lado, a simplicidade faz deste protocolo um protocolo com baixa taxa de transmissão, baixa fiabilidade e sem segurança. A taxa de transmissão efectiva máxima é de 60bps (um bit por cada ciclo de rede) o que se torna insuficiente para trocas de dados significativas entre nós na rede. A lacuna de segurança e fiabilidade afastam o uso do X10 em aplicações domésticas críticas, tais como o controlo de uma porta de entrada de uma casa [8].

1.2.1.2. HomePlug

*HomePlug Powerline Alliance*³ é um consórcio industrial criado com o objectivo de uniformizar as comunicações de alto ritmo de transmissão de uso doméstico, tais como dados, voz e vídeo. Entre estas, estão definidas duas especificações, a *HomePlug 1.0* e a *HomePlug AV*.

Testado por fornecedores de serviços e fabricantes de produtos electrónicos, e graças à uniformização, o *HomePlug AV* permite acesso instantâneo e ligação na mesma rede de serviços como Internet de banda larga, TV, IPTV, vídeo (tanto standard como de alta definição), streaming, etc. Com velocidades até 200Mbps, está ainda concebido para fornecer uma comunicação robusta sobre as linhas de alimentação que satisfaz as regulamentações a nível mundial. Esta tecnologia contorna o problema de interferências das linhas com soluções no domínio temporal e no domínio da frequência [9].

O espectro da tecnologia *HomePlug AV* varia entre os 1,8 e os 30MHz, uma banda muito acima de outros sinais, possivelmente existentes na mesma linha, tais como o X10. Este facto permite isolar o espectro *HomePlug AV* de sinais de baixa frequência com o auxílio de filtros passivos. No entanto, os filtros passa-alto não

³ <http://www.homeplug.org/>

previnem a interferência de ruído introduzido, por exemplo, por electrodomésticos. Assim, a fim de contornar este problema, a camada física⁴ (PHY) deste protocolo usa o esquema de modulação em banda-larga, OFDM - Orthogonal Frequency Division Multiplexing.

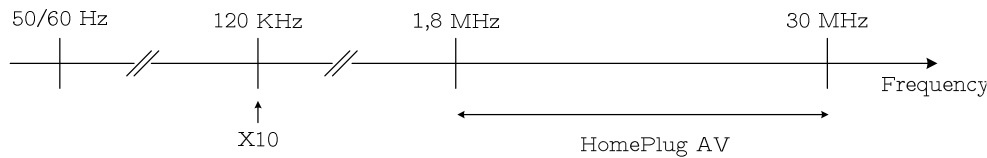


Fig. 5 – Espectro de frequência dos protocolos X10 e HomePlug AV

O protocolo MAC (Medium Access Controller), na tecnologia HomePlug AV, é uma variante do bem conhecido e tão usado CSMA/CA - Carrier Sense Multiple Access with Collision Avoidance, ao qual foram adicionados alguns aperfeiçoamentos para permitir o suporte de prioridades, critérios de justiça e permitir o controlo da latência [10].

O HomePlug AV proporciona ainda uma poderosa tecnologia de codificação que permite uma eficiente correcção de erros que, na maioria dos casos, evita a necessidade de reenvio do pacote de dados por parte do transmissor sendo mesmo possível reconstruir o pacote de dados original em tempo real, o que permite um elevado *throughput*⁵ [9].

Existem dois tipos de frames na tecnologia HomePlug, a Long Frame e a Short Frame. A Long Frame consiste no campo SOF, campo de dados (payload) e o campo End Of Frame (EOF). A Short Frame consiste delimitador de resposta (Response Delimiter) e é usada como parte do processo de pedido automático de reenvio de pacotes corrompidos. Este processo tem o nome de Automatic Repeat Request (ARQ) e ajuda a reduzir o número de pacotes perdidos.

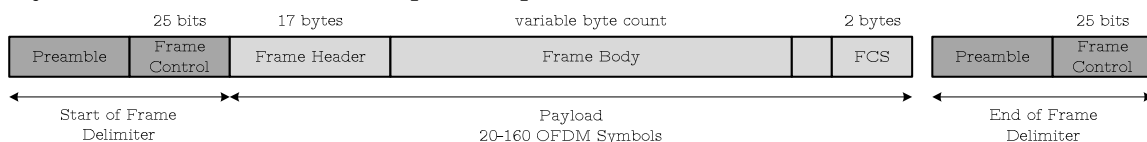


Fig. 6 – Long Frame – HomePlug

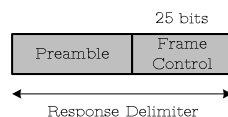


Fig. 7 – Short Frame – HomePlug

Sendo uma tecnologia poderosa, com fortes garantias de segurança e privacidade, ao mesmo tempo que permite uma vasta interacção com os mais variados

⁴ A camada física (PHY) é a primeira camada do modelo ISO/OSI, que define a voltagem, codificação, modulação, velocidade de barramentos e pinos.

⁵ Em comunicações, o *throughput* é a quantidade de pacotes de dados que passam através de um canal de comunicação e são correctamente entregues ao destinatário, por unidade de tempo.

tipos de redes domésticas, a rede HomePlug sobre Powerline torna-se a espinha dorsal das redes de comunicação doméstica de alta velocidade.

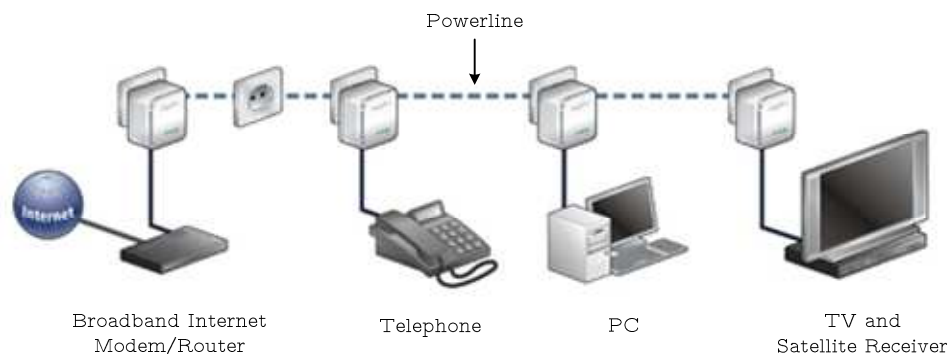


Fig. 8 – Implementação HomePlug AV num ambiente doméstico [11]

1.2.2. A solução PLC para DC e a tecnologia YAMAR

Como visto anteriormente a tecnologia Powerline é já usada, com bastante conhecimento e segurança, em ambiente doméstico. No entanto, existe um grande interesse na possibilidade de introduzir esta solução em veículos, onde é necessário introduzir um sistema completamente compatível com os sistemas actualmente existentes, nomeadamente o barramento CAN que é o barramento mais usado actualmente nos automóveis.

Uma solução para o domínio automóvel obriga ao funcionamento da comunicação Powerline sobre linhas de alimentação de baixa voltagem DC, devido às condições impostas pela alimentação a baterias (12 ou 42V) e restantes componentes electrónicos constituintes do automóvel, que são alimentados a, por exemplo 3.3, 5 ou 9V, o que implica uma tecnologia de acoplamento diferente.

Uma implementação direccionada para o ramo automóvel terá que preencher vários requisitos, de forma a ser compatível com a tecnologia já existente neste campo. A implementação PLC terá que se ajustar aos componentes e protocolos já existentes para uma fácil integração nos sistemas automóveis, sem que seja necessário efectuar grandes mudanças de tecnologias e infra-estruturas. Estando assim aberto o caminho para uma maior aceitação por parte da industria automóvel [12].

Existem assim dois cenários a explorar:

- i. Fornecer um sistema de comunicação diferente, em concordância com as tecnologias actuais usadas nos sistemas automóveis, tendo que fornecer a largura de banda e sensibilidade adequadas e tolerância a interferências electromagnéticas (EMI).
- ii. Fornecer um sistema que substitua fisicamente os sistemas existentes e que seja transparente ao nível da aplicação. Neste caso, a tecnologia PLC deve fornecer um comportamento temporal, semelhante, ou igual, ao barramento original.

Há algumas questões que terão que ser contabilizadas no momento da escolha entre esta e outras tecnologias já existentes. A questão principal prende-se com o facto

da menor eficiência da tecnologia PLC em relação às tecnologias mais poderosas existentes, como é o caso do FlexRay. No entanto, este problema verifica-se apenas no domínio automóvel, pois para o domínio da domótica a tecnologia PLC já ultrapassa os protocolos existentes no campo automóvel, em termos de largura de banda. Por outro lado, a largura de banda da PLC é comparável à oferecida por LIN ou CAN.

Outra questão que merece especial atenção é a segmentação da rede, pois a tecnologia PLC usa transmissão broadcast (todos os nós recebem o que cada um transmite), ao passo que as tecnologias existentes possuem segmentos isolados. Esta questão será um possível inconveniente, visto que conduzirá à necessidade de desenvolvimento de soluções especiais para particionar a rede PLC em sub-redes. Este particionamento é essencial para garantir isolamento e fácil integração entre redes com necessidades e características bastante distintas, por exemplo, a rede do sistema de travagem e a rede de controlo dos vidros eléctricos no automóvel. Não é desejável que estas duas redes distintas recebam informação que não lhes é útil nem desejada.

Nos sistemas típicos, já existentes, este nível de isolamento e integração é garantido pela introdução de gateways, que não é mais que uma ECU especial, ligada a várias sub-redes, que filtra o tráfego, deixando passar apenas aquele que é destinado a cada uma das ECUs.

Nos automóveis, habitualmente, existe apenas uma fonte de alimentação para todos os componentes, a bateria. Assim, numa solução PLC, isto equivale a partilhar o barramento por todos os componentes existentes num automóvel, o que pode não ser desejável. Tendo apenas um barramento para todos os componentes pode alcançar-se rapidamente o número máximo de nós suportados por esse mesmo barramento e, por outro lado, esta rede única fica assim muito sensível a interferências geradas por todos os componentes, que devem assim ser isoladas para que cada nó não afecte os restantes.

A solução para este problema passa pela colocação de filtros que sejam capazes de deixar passar apenas a alimentação para outros nós ao mesmo tempo que bloqueiam os dados, os quais devem ser encaminhados para o(s) nó(s) pretendidos através de um gateway. Outra solução, menos prática, seria a utilização de baterias independentes, o que levaria a um aumento dos custos e a problemas de espaço.

Ao usar a PLC como uma substituição física de um protocolo já existente, como, por exemplo, CAN, há um aspecto muito importante a ter em consideração, a camada MAC⁶. Esta tem que cumprir as mesmas especificações de construção de forma a garantir os mesmos tempos de resposta e ordem das mensagens enviadas da tecnologia a substituir. Pois, caso apenas se verifique uma destas propriedades, cumprir o tempo de resposta mas não cumprir a ordem das mensagens ou vice-versa, o sistema terá uma resposta não desejada, diferente da resposta do sistema original.

Por outro lado, o sistema PLC pode ser usado como um sub-sistema paralelo (redundante) a sub-sistemas já existentes, tendo como função suportar o funcionamento do sistema geral em caso de falha do sub-sistema original.

Esta tecnologia pode ser aplicada nos veículos actuais sem que seja necessário efectuar mudanças nos protocolos de comunicação implementados. Esta vantagem é

⁶ A camada MAC é uma sub-camada da Data Link Layer (DLL) especificada no modelo OSI de sete camadas. Controla os endereços e acessos ao canal de comunicação permitindo assim comunicações multiponto. A camada DLL é a número dois das sete existentes no modelo OSI.

conseguida com a introdução de circuitos (des)moduladores entre os módulos existentes e o barramento DC que funcionam como um conversor autónomo. Para novas implementações (novos modelos de veículos e novos subsistemas) a implementação do conversor pode ser feita no novo circuito, simplificando assim o sistema como um todo e a sua instalação. Este circuito será assim responsável por transmitir e receber mensagens para e do módulo de destino.

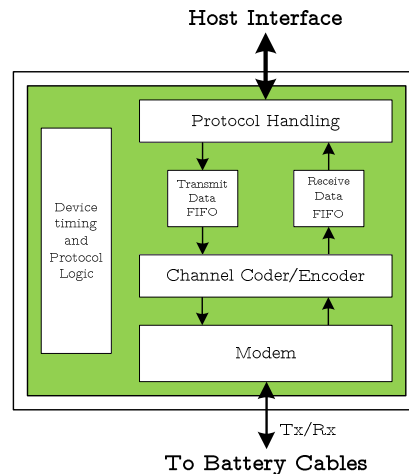


Fig. 9 – Circuito de interligação CAN, DC-Bus – Modem DC-CAN

Como tecnologia de (des)modulação pode usar-se a técnica PSK (Phase Shift Keying), na qual a fase da portadora é modificada de acordo com o sinal a ser transmitido, sendo assim possível enviar sinais de dados na linha de alimentação de potência. Esta tecnologia tem uma elevada performance e elevada imunidade ao ruído.

Este circuito electrónico (modulador/desmodulador) permite a configuração de qualquer frequência de operação possibilitando assim o uso de vários canais sobre a mesma linha [13].

Uma das soluções mais promissoras, actualmente no mercado, é a tecnologia da empresa Israelita YAMAR Electronics que apresenta neste momento modems transdutores DC-BUS a operar sobre linhas de alimentação DC, com ruído, destinadas à indústria automóvel, aviação, aeroespacial e redes industriais multiplexadas, que será explorada com mais detalhe no Capítulo 2, Secção 2.4.

1.3. Objectivo da Tese

Com o aparecimento de novas tecnologias e melhoramento das já existentes, os sistemas electrónicos utilizados actualmente no domínio automóvel são cada vez mais complexos. Desde há vários anos que essa complexidade resultou na utilização extensiva de arquitecturas distribuídas, recorrendo a redes específicas como CAN, LIN, TTP/C e mais recentemente FlexRay, falando-se agora também de Ethernet. Contudo, um dos problemas que sempre esteve presente nestes sistemas é o da respectiva cablagem, cujo peso e instalação trazem problemas acrescidos de flexibilidade de projecto e de evolução, bem como de fiabilidade.

Uma tecnologia que tem sofrido uma grande evolução recentemente é a tecnologia Powerline, em particular para viaturas, que utiliza as linhas de alimentação de potência dos nós para transferir informação. A utilização desta tecnologia permitirá reduzir significativamente a cablagem, aliviando os problemas a ela associados.

Assim, o propósito principal desta tese é efectuar a caracterização do canal de comunicação PLC, para averiguar a sua viabilidade como sistema substituto da tão usada rede CAN, nos automóveis, focando no comportamento temporal.

Começou-se por efectuar alguns testes de comunicação baseados em nós CAN com o intuito de ambientação ao processo de medida de características do canal de comunicação, tais como throughput e tempos de resposta.

A segunda parte deste trabalho foi dedicada ao estudo das propriedades temporais do canal de comunicação PL, baseado nos módulos DCB500, desenvolvidos pela empresa Israelita YAMAR Electronics Lda.⁷ que apresentam ser uma alternativa promissora como tecnologia substituta da actualmente existente.

1.4. Parceiros

Este trabalho encontra-se integrado num projecto de investigação em curso, no qual estão envolvidas as seguintes instituições:

- Universidade de Aveiro – Portugal,
- Università degli studi di Pavia – Itália e
- Mälardalen Real-Time Research Centre (MRTC), Mälardalen University – Suécia.

Estas parcerias permitiram que a realização da segunda parte deste trabalho fosse efectuada nas instalações da Università degli studi di Pavia – Itália, durante três meses.

1.5. Estrutura da Tese

No capítulo 1, INTRODUÇÃO, foi abordada uma visão geral sobre as comunicações no domínio automóvel, a alternativa da comunicação Powerline, explorando as soluções existentes no domínio doméstico sobre linhas AC e a solução PLC para DC focando a tecnologia YAMAR.

No capítulo 2, REDES UTILIZADAS NO DOMÍNIO AUTOMÓVEL, tal como o próprio título indica, irão ser abordados os protocolos de comunicação existentes actualmente no domínio automóvel, LIN, CAN, FlexRay e finalmente a tecnologia DC-Powerline Yamar.

O capítulo 3, CARACTERIZAÇÃO TEMPORAL DO CANAL DE COMUNICAÇÃO CAN, irá apresentar os resultados obtidos na caracterização do canal de comunicação CAN.

⁷ <http://www.yamar.com/index.html>

No capítulo 4, CARACTERIZAÇÃO TEMPORAL DO CANAL DE COMUNICAÇÃO POWERLINE, irão ser apresentadas as medidas, esquemas e resultados da caracterização temporal do canal de comunicação Powerline.

Por fim, no capítulo 5, CONCLUSÕES E TRABALHO FUTURO, irão ser percorridas algumas ideias e conclusões acerca do trabalho efectuado e de algum do muito trabalho que haverá ainda para efectuar nesta área.

2. REDES UTILIZADAS NO DOMÍNIO AUTOMÓVEL

Com o evoluir da tecnologia e com o aumento da electrónica nos automóveis, tornou-se imperativo desenvolver protocolos de comunicação entre os vários subsistemas, permitindo assim reduzir a quantidade de cablagem e facilitar a interacção e ligação de dispositivos de diferentes fabricantes, aumentando também a qualidade e fiabilidade das comunicações.

Assim, os protocolos de comunicação no domínio automóvel estão actualmente uniformizados em grande medida, e dentro dos existentes destacam-se os apresentados de seguida, isto é, CAN, LIN e FlexRay.

2.1. CAN

O protocolo CAN - Controller Area Network é um protocolo de comunicação digital série que suporta de uma forma muito eficiente o controlo de sistemas distribuídos de tempo-real com um nível muito elevado de segurança e fiabilidade [14].

Foi desenvolvido, em finais dos anos 80, por uma das maiores sociedades industriais privadas a nível mundial, a BOSCH, como um barramento multi-master, do tipo broadcast com um ritmo de transmissão máximo de 1Mbps, com especial propósito de obter um protocolo que permitisse uma rápida troca de dados entre controladores electrónicos em veículos motorizados. O principal objectivo era construir automóveis mais fiáveis, seguros e eficientes, diminuindo simultaneamente o peso e a complexidade das cablagens. Desde a sua criação, o protocolo CAN tem ganho cada vez mais popularidade no domínio automóvel e automação industrial. Paralelamente, outros mercados em que o uso de redes de comunicação pode trazer grandes benefícios, tais como equipamentos médicos ou equipamentos móveis, começam também a usar o protocolo CAN [15].

Ao contrário dos protocolos de comunicação tradicionais, como o USB ou Ethernet, CAN não tem a capacidade de enviar grandes blocos de dados ponto-a-ponto, de um nó A para um nó B. Numa rede CAN são trocadas mensagens curtas com informação como temperatura ou as RPM (Rotações Por Minuto) do motor de um automóvel, para toda a rede, ou seja, todos os nós do barramento recebem todas as transmissões efectuadas. Não existe, assim, uma forma de enviar uma mensagem apenas para um nó específico. Contudo, o controlador de rede CAN dispõe de um filtro local que lhe permite filtrar apenas as mensagens que a ele são destinadas.

Usando apenas um par de condutores entrançados como linha de dados série, este protocolo possui mecanismos que permitem o seu funcionamento em ambientes electricamente ruidosos, é de fácil instalação e modificação e possui ainda mecanismos de detecção automática e correcção de erros, CSMA/BA, isto é, cada nó na rede tem que monitorizar o barramento e detectar um período de não actividade, antes de tentar transmitir (CS – Carrier Sense). Por outro lado, quando é detectado um período de não actividade no barramento, todos os nós têm igual oportunidade para transmitir (MA – Multiple Access). Se dois nós começarem a transmitir ao mesmo tempo, cada um deles irá detectar a colisão e tomar a posição adequada, avançar ou abortar a sua transmissão, dependendo do seu nível de prioridade, definido bit a bit (BA – Bitwise Arbitration).

Em CAN é usado um método de arbitragem não destrutivo, isto é, quando dois ou mais nós tentam enviar bits distintos o resultado corresponde a um dos bits, o

chamado dominante, o qual se sobrepõe ao outro, chamado recessivo. Assim, todo o processo de arbitragem é efectuado sem corrupção ou atrasos da mensagem de maior prioridade, a qual corresponde à que tem um identificador com bits dominantes nos bits mais significativos.

O barramento usa o código NRZ (Non-Return to Zero) com bit-stuffing. Se um dos nós ligados ao barramento estiver a forçar um bit dominante ("0") na linha, toda a linha estará a zero, independentemente dos nós que estiverem a tentar transmitir um bit recessivo ("1"). O uso de bit-stuffing deve-se ao facto da necessidade de manter a sincronização de relógio, isto é, quando um emissor detecta 5 bits consecutivos de igual polaridade, cinco bits a "1" ou cinco bits a "0", ele insere automaticamente um bit complementar na mensagem a ser transmitida para forçar transições de nível no barramento. Estes stuff bits são também automaticamente removidos pelo receptor.

2.1.1. Mensagens CAN

CAN usa mensagens curtas que não dispõem de um endereço explícito, no entanto pode afirmar-se que as mensagens são endereçadas ao conteúdo, isto é, o seu conteúdo define, implicitamente, o seu endereço através de um identificador (ID) único.

Cada mensagem pode transmitir de 0 a 8 bytes de informação (payload). Contudo, é possível transmitir dados de maior tamanho efectuando a sua segmentação por várias mensagens.

A rede CAN pode ser configurada para funcionar com dois tipos de mensagens diferentes, o formato base ou standard (CAN 2.0A) ou o formato estendido (extended format – CAN 2.0B). A diferença entre estes dois tipos está no ID da mensagem. A mensagem standard possui um ID de 11 bits e a mensagem "extended" possui um identificador de 29 bits, sendo os 18 bits como uma extensão do identificador standard. Esta distinção é feita usando um bit, o bit IDE, da mensagem CAN, que é enviado como dominante ("0") no caso de uma mensagem standard, de identificador de 11 bits, e como recessivo ("1"), no caso de uma mensagem de identificador de 29 bits.

Os controladores CAN que suportam CAN 2.0B estão também habilitados a enviar e receber mensagens no formato CAN 2.0A. Todas as mensagens começam com um bit que indica o início da transmissão de uma mensagem, o "start-of-frame" bit [16].

O protocolo disponibiliza quatro tipos de mensagens (*frames*) que são a *Data Frame*, *Remote Frame*, *Error Frame* e *Overload Frame*.

2.1.1.1. Data Frame

"Hello everyone, here's some data labeled X, hope you like it!"

A Data Frame é a mensagem mais comum na comunicação CAN. É a mensagem que transporta os dados.

Este tipo de mensagem é constituído pelo campo de arbitragem (Arbitration Field) que define quem ganha o acesso ao barramento quando dois ou mais nós competem pela sua posse, o campo de dados (Data Field) que contém entre 0 e 8

bytes de dados a serem enviados, o campo CRC (CRC Field) que contém 15 bits de verificação usados para detecção de erros nas mensagens, o campo de confirmação de mensagem recebida (Acknowledgement Field) que indica se algum nó recebeu ou não a mensagem transmitida e por fim o campo que indica o final da mensagem (End Of Frame).

O campo de arbitragem contém ainda algumas diferenças consoante a versão da Data Frame (2.0A ou 2.0B). Para a versão 2.0A, é constituído pelo identificador de 11 bits e o bit RTR que é enviado como dominante no caso das Data Frames. Para a versão 2.0B o Arbitration Field é dividido em duas secções, o identificador base, 11 bits, e a extensão do identificador, 18 bits, que compõe assim o identificador de 29 bits. A separar estas duas secções estão dois bits, o SRR e o IDE. O bit SRR é um substituto do RTR e é enviado como recessivo, sendo assim o bit IDE que define se a mensagem enviada é uma Standard Frame ou uma Extended Frame.



Fig. 10 – Data Frame, identificador de 11 bits



Fig. 11 – Data Frame, identificador de 29 bits

2.1.1.2. Remote Frame

"Hello everyone, can somebody please produce the data labeled X?"

A Remote Frame é semelhante à Data Frame, com a excepção da não existência do Data Field e esta estar explicitamente identificada como Remote Frame através do bit RTR que, neste caso, é enviado como recessivo.

Esta mensagem tem como objectivo permitir solicitar a outro nó a transmissão de uma mensagem. Se, por exemplo, um nó A enviar uma Remote Frame com o campo de arbitragem 701, um nó B, desde que devidamente inicializado, poderá então responder com uma Data Frame com Arbitration Field também a 701.

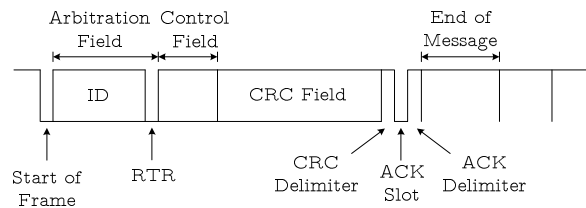


Fig. 12 – Remote Frame, do tipo CAN 2.0A

2.1.1.3. Error Frame

(everyone, aloud) "OH DEAR, LET'S TRY AGAIN"

Uma mensagem de erro é uma mensagem que viola as regras de construção de Frames CAN. Este tipo de mensagem é enviado por um nó quando detecta erros numa mensagem recebida, fazendo com que os outros nós enviem também mensagens de erro. O transmissor que enviou a mensagem recebida com erro tenta então enviá-la de novo.

O barramento CAN possui um contador de erros que permite salvaguardar que um nó não destrói a comunicação devido a repetidos envios de mensagens de erro.

Estas mensagens consistem numa "Error Flag", que é constituída por seis bits do mesmo valor, violando assim a regra de bit-stuffing, e um "Error Delimiter" formado por oito bits recessivos.

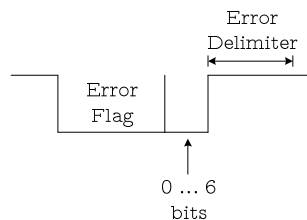


Fig. 13 – Error Frame

2.1.1.4. Overload Frame

"I'm a very busy little microprocessor, could you please wait for a moment?"

Este tipo de mensagem é semelhante à "Error Frame" e é enviada por um nó quando este se encontra demasiado ocupado e não pode receber uma mensagem. No entanto, o envio de uma "Overload Frame" não incrementa o contador de erros e não despoleta um reenvio de mensagem por parte do transmissor. Cada nó poderá transmitir apenas duas "Overload Frames" consecutivas.

Contudo, com a evolução da tecnologia, os microprocessadores actuais, provavelmente, nunca enviarão este tipo de frame, visto que são suficientemente poderosos para contornar este problema [17].

2.1.2. O processo de arbitragem

O processo de arbitragem (que define que nó ganha o acesso ao barramento) é extremamente importante para a gestão da largura de banda real disponível para transmissão de dados.

Qualquer controlador CAN pode iniciar a transmissão quando detecta inactividade no barramento. Assim, podem existir vários controladores a iniciar a transmissão simultaneamente e a questão da arbitragem é resolvida da seguinte forma. Cada nó que está a transmitir está também a monitorizar o barramento bit a bit. Se o nó detectar um nível dominante no barramento enquanto está a transmitir um bit recessivo abandona imediatamente o processo de arbitragem e torna-se nó receptor outra vez. Este processo efectua-se até ao final do campo de arbitragem e quando este é completamente enviado garante-se que apenas um dos nós ficou com a posse do barramento, que manterá, até ao final da transmissão da sua mensagem. Os outros nós irão tentar retransmitir as suas mensagens quando o barramento estiver novamente disponível. Não é, assim, desperdiçado tempo no processo de arbitragem.

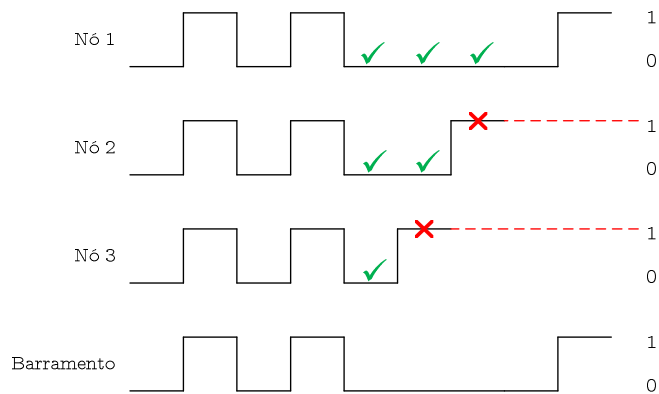


Fig. 14 – Processo de arbitragem de acesso ao barramento CAN

Uma importante condição, para que este processo ocorra com sucesso, é garantir que não existem dois ou mais nós com o mesmo campo de arbitragem (ID da mensagem). As mensagens de cada nó têm um ID próprio e único no sistema.

2.2. LIN

O protocolo LIN é um protocolo de baixa velocidade e baixo custo para redes automóveis que surge como sub-rede do barramento CAN para integrar sensores inteligentes ou actuadores nos automóveis, onde a largura de banda e versatilidade de CAN não são requeridas. Permite, assim, a implementação de uma rede veicular hierarquizada, reduzindo os custos e melhorando a qualidade dos veículos [18].

É um protocolo de comunicação série com suporte eficiente de controlo de nós mecatrónicos⁸ em aplicações de redes distribuídas. E tem como principais características:

- Conceito de um único Master com múltiplos Slaves.
- Tamanho da frame de dados variável (2, 4 e 8 bytes).
- Algoritmo de detecção e correcção de erros.
- Auto-sincronização, sem oscilador de quartzo ou cerâmica nos nós Slave.
- Transmissão determinística do sinal com cálculo do tempo de propagação do sinal com antecedência.
- Implementação de baixo custo devido à comunicação feita através de um único fio.
- Velocidade até 20kbps com um máximo de 40m de comprimento do barramento.
- Voltagem de operação de 12V

2.2.1. Mensagens LIN

Todos os dados são enviados numa mensagem que contém um cabeçalho (*Header*), uma resposta (*Response*) e algum tempo entre eles para que o nó Slave tenha tempo para responder. Cada frame é enviada numa *frame slot* determinada anteriormente pelo programador do software do nó Master.

As mensagens são criadas quando o nó Master envia uma frame contendo o Header e de seguida o(s) nó(s) Slave preenche(m) a frame com dados de acordo com o cabeçalho enviado pelo Master [19].

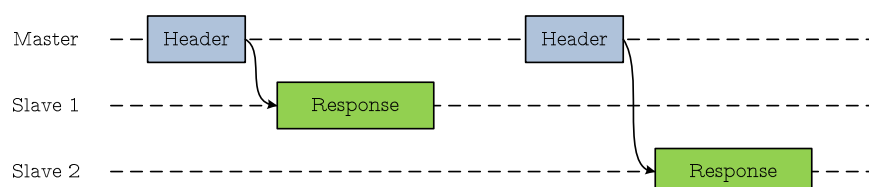


Fig. 15 – Exemplo de comunicação LIN com dois nós

O *Header* é constituído por três partes, *Break*, *Sync* e *Identifier*. O campo *Break* é uma ausência de comunicação equivalente ao tempo de 13 bits e serve para alertar todos os Slaves que devem ficar à espera dos restantes campos. O campo *Sync* consiste num byte de dados de valor 0x55 que será usado pelos Slaves para medir o tempo de bit no barramento e recalculer o seu baud rate interno, mantendo-os assim sincronizados. O *Identifier* define uma acção a ser executada por um ou mais Slaves. Se o identificador provocar que um nó Slave envie uma resposta, isto é, que este envie dados para o barramento, o *Identifier* é chamado *Rx-Identifier*. Caso seja o Master a enviar dados através do barramento o identificador é chamado de *Tx-Identifier*.

O campo *Response* está dividido nos campos *Data* e *Checksum*. No campo *Data* o Slave que responde pode enviar de zero a oito bytes de dados para o barramento. O

⁸ **Mecatrónica** é a conjugação de **Mecânica** e **Electrónica**. É a combinação da engenharia mecânica, electrónica, controlo, design de sistemas e engenharia de computadores para criar novos e úteis produtos.

tamanho do campo de dados é definido pelo projectista da aplicação. Existem dois modelos de Checksum. O primeiro, usado até à versão 1.3, inclui apenas os bytes de dados, o segundo modelo inclui o campo de dados e o identificador.

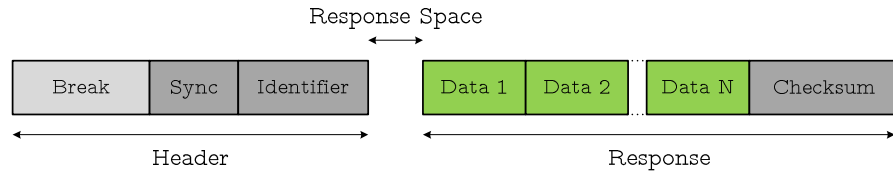


Fig. 16 – Frame LIN

2.2.1.1. Unconditional Frame

Este tipo de Frame é a mais usual na comunicação LIN. O nó Master envia o Header na respectiva *frame slot* e o nó Slave designado preenche a frame com dados.

2.2.1.2. Event-Triggered Frame

O objectivo deste método é receber a maior quantidade de dados dos nós Slave sem ocupar demasiada largura de banda. Uma frame *event-triggered* pode ser preenchida com dados de mais do que um nó Slave. Um Slave só actualiza os dados na frame se o seu valor tiver sido alterado. Se mais que um Slave quiser actualizar os dados na mensagem ocorre uma colisão. Neste caso, o Master deve então enviar *Unconditional Frames* para cada um dos Slaves começando pelo de maior prioridade.

2.2.1.3. Sporadic Frame

Este tipo de mensagens permite imprimir algum dinamismo ao protocolo LIN, que por si só é um protocolo estático. O cabeçalho de uma mensagem deste tipo, *Sporadic Frame*, apenas é enviado pelo Master quando este sabe que um sinal foi actualizado num nó Slave. Normalmente é o Master que preenche o campo de dados da frame e os nós Slave são receptores dessa informação [19].

2.2.1.4. Diagnostic Frame

Uma mensagem de diagnóstico transporta sempre dados de diagnóstico ou de configuração e contém sempre 8 bytes de dados.

O protocolo LIN é assim, um protocolo de fácil utilização, mais económico que CAN e outros protocolos de comunicação, permite a redução de cablagem, contribui para veículos mais fiáveis, é de fácil expansão e não exige o pagamento de licenças.

Assim, LIN não é uma solução que permita substituir completamente o barramento CAN, mas é uma excelente alternativa quando os custos são um factor mais importante que a velocidade e largura de banda [2].

2.3. FlexRay

O FlexRay é um protocolo de comunicação série de alta velocidade para redes veiculares e é uma extensão do protocolo Byteflight⁹. Preenche todos os requisitos de performance para segurança activa, tais como canais de transmissão redundantes e mecanismos de tolerância a falhas. Entre as aplicações do FlexRay estão *steer-by-wire* e *brake-by-wire* [20], que são sistemas de direcção e travagem, respectivamente, em que os componentes mecânicos e/ou hidráulicos de ligação são substituídos por barramentos de comunicação, neste caso, o barramento FlexRay. Estes sistemas são genericamente conhecidos por *X-by-wire*.

O FlexRay é ideal para este tipo de sistemas devido às suas características de comunicação *time-triggered* com características determinísticas, tolerância a falhas e elevadas taxas de transmissão [21], a sua largura de banda máxima de 10Mbps e a possibilidade de dois canais de comunicação para redundância [22].

O FlexRay utiliza várias ECU's, mas apenas uma comunica de cada vez, havendo assim uma ECU emissora e uma ECU receptora, os quais comunicam entre si através de um barramento. Como descrito anteriormente, a topologia de grupo de comunicação do protocolo FlexRay permite ter um canal (*Single Channel*) ou dois canais (*Dual Channel*) [23] e um nó da rede pode conectar-se a ambos os canais, Fig. 17 – nós A, B e E, ou a um só canal tal como é o caso dos nós C ou D. Para a implementação *Single Channel* todos os nós da rede são conectados a um só canal, Canal 1 ou 2.

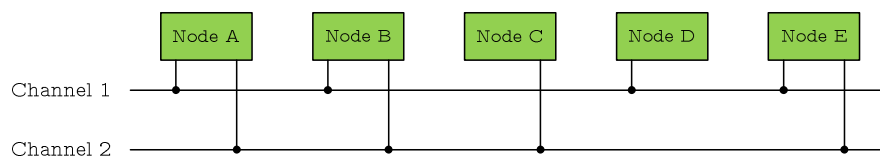


Fig. 17 – Configuração Dual Channel – FlexRay

Em cada instante, apenas uma ECU acede ao barramento. Cada bit a ser enviado é mantido no barramento por 8 ciclos de relógio, o receptor guarda num buffer as últimas 5 amostras e usa a sua moda¹⁰ como sinal de entrada.

O protocolo FlexRay tem duas fases, a primeira das quais, chamada estática funciona de acordo com o princípio TDMA - Time Division Multiple Access, em que cada nó tem *slots* de tempo predefinidas nas quais tem acesso exclusivo ao barramento. Estas *slots* são repetidas num ciclo temporal definido. Assim, o tempo que cada mensagem está no barramento pode ser exactamente previsto e o acesso ao barramento é determinístico [24]. A segunda parte, chamada dinâmica, funciona de acordo com o protocolo Byteflight e usa um mecanismo de arbitragem do tipo *mini-slotting*.

⁹ <http://www.byteflight.com/>

¹⁰ **Moda** (estatística) - valor mais frequentemente representado numa série de observações.

2.3.1. A ECU FlexRay

A ECU FlexRay consiste num processador (*Host Processor*), um Controlador de Comunicação (CC), dois *Bus Guardian* (BG) e dois *Bus Drivers* (BD). O processador fornece e processa os dados que serão enviados através do controlador FlexRay [24].

O *Bus Guardian* monitoriza o acesso ao barramento e o processador informa-o das slots de tempo alocadas pelo controlador de comunicação. O *Bus Guardian* permite então que o Controlador de Comunicação transmita os dados apenas na sua slot e activa o *Bus Driver*.

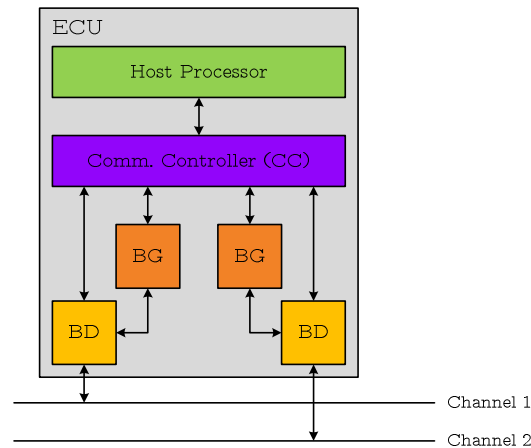


Fig. 18 – Exemplo de uma Electronic Control Unit FlexRay

2.3.2. A Frame FlexRay

A frame FlexRay está dividida em três segmentos, o cabeçalho (*Header*), o campo de dados (*Payload*) e o *Trailer*¹¹.

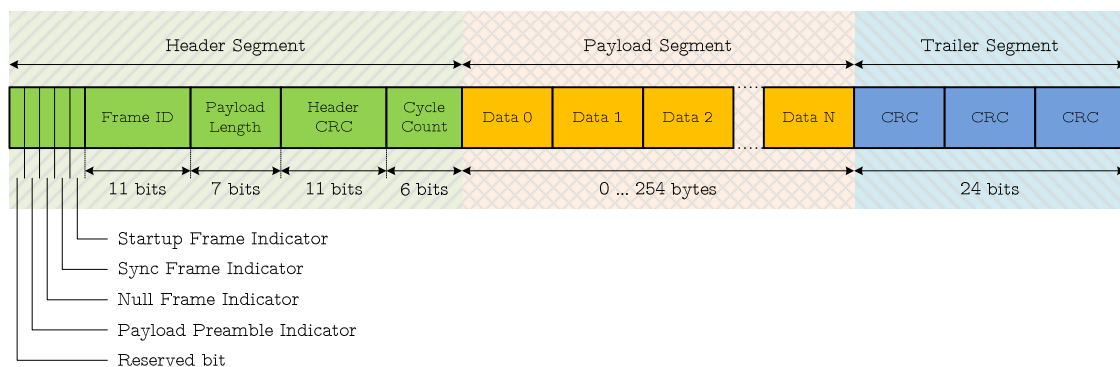


Fig. 19 – Frame FlexRay

Como o FlexRay é um sistema determinístico há consequências na forma como estes segmentos são transmitidos na rede. Como mostra a Fig. 19, o cabeçalho aparece

¹¹ **Trailer** é uma quantidade de dados complementar, colocado no final de um bloco de dados que estão a ser transmitidos ou armazenados. Pode conter informação para facilitar o tratamento dos dados ou simplesmente para marcar o seu fim.

em primeiro lugar, seguido pelo segmento de dados e por fim o segmento *Trailer*, que é o último a ser transmitido [25].

2.3.2.1. Header

Protegido por um código extra de CRC, o cabeçalho contém informação do controlo do protocolo como flag de sincronização da frame, o ID da slot a que pertence a frame, o tamanho da frame e um contador de ciclos.

2.3.2.2. Payload

Este segmento contém entre 0 e 254 bytes. Caso este segmento não contenha quaisquer dados, o indicador de frame nula (*null frame indicator*) é colocado a 1. Opcionalmente a secção de dados pode conter um ID da mensagem, o qual identifica que tipo de dados é transportado pela frame. Os bytes deste segmento são identificados numericamente, começando em 0 para o primeiro byte depois do segmento Header e incrementando de uma unidade a cada byte posterior [25].

2.3.2.3. Trailer

O segmento *Trailer* contém um CRC de 3 bytes, para proteger toda a frame e possibilitar a detecção de erros. O campo CRC é calculado sobre os segmentos *Header* e *Payload*. O cálculo é feito por hardware e inclui todos os campos destes segmentos.

A rede de comunicação FlexRay abrange comunicação determinística, tolerância a falhas e barramento de alta velocidade, cumprindo assim as especificações para as próximas gerações de automóveis [26].

2.4. A tecnologia DC-Powerline Yamar

Fundada em 1994, a YAMAR Electronics desenvolve soluções de comunicação sobre linhas de alimentação com ruído, isto é, linhas de alimentação de potência com aplicação nos mais variados campos, tais como o campo veicular, a aviação, a electrónica de consumo, segurança, indústria aeroespacial e brinquedos.

A tecnologia DC-Bus possibilita a construção de redes de comunicação de dados, voz, música e vídeo através das linhas de alimentação DC tais como as existentes em todos os automóveis. Esta tecnologia funciona sobre ambientes adversos e com ruído tal como as linhas de alimentação das baterias e a mesma linha de alimentação pode comportar múltiplas redes independentes usando, para tal, diferentes frequências da portadora.

Como soluções para o campo automóvel, a Yamar Electronics disponibiliza produtos compatíveis com os protocolos CAN-Bus, LIN sub-bus e J1850 bus.

No campo da mecatrónica dispõe de soluções para comunicação assíncrona com taxas de transferência de dados até 60Kbps para controlo de aplicações em sub-barramentos, como é o caso de LIN. Esta solução permite incrementar a velocidade de transmissão de LIN, reduzindo os custos dos fios de controlo e transdutores LIN ou

RS232, operando ainda sobre linhas até 42V. Estes produtos para comunicação multiplexada a 10Kbps, da família QDC, são uma solução flexível para aplicações de comando e controlo como é o caso de sensores e actuadores tão frequentemente usados nos veículos, como ar condicionado, módulos das portas, detectores de obstáculos, etc [27].

No campo da telemática, a Yamar oferece dois produtos, o DCB500 e DCAN250. O DCB500, que é o objecto de estudo desta tese, foi desenhado para comunicações até 500Kbps de grandes quantidades de dados, por exemplo, voz. O DCAN250 possui interfaces com controladores CAN-Bus. As placas de avaliação (Evaluation Boards – EB) destes produtos encontram actualmente sob testes pelos principais fabricantes de automóveis. A Evaluation Board (EVB) DCB500 encontra-se também sob testes no projecto de investigação no qual se insere esta dissertação.

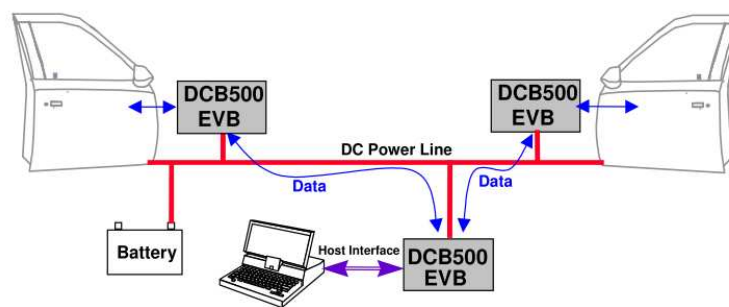


Fig. 20 – Exemplo de configuração de rede baseada em Evaluation Boards DCB500 [28]

No campo multimédia, para comunicações de vídeo e música, é oferecido o produto DCB101. A Yamar desenvolveu um protótipo para transmissões com qualidade de CD, ou seja, 1,7Mbps sobre as linhas de alimentação. Uma possível aplicação será a conexão entre a caixa de CD's instalada na traseira do automóvel e unidade de controlo, rádio, instalado no painel de instrumentos [27].

De seguida são apresentadas em mais detalhe as principais soluções disponibilizadas pela Yamar Electronics, SIG40, DCAN250 e DCB500, com especial incidência na Evaluation Board DCB500, que é o produto em estudo neste trabalho.

2.4.1. SIG40

O modem transdutor SIG40 permite a construção de redes de comunicação UART/LIN assíncronas operando sobre linhas de alimentação, usando as mesmas tecnologias do sinal original.

Quando aplicado num barramento LIN sobre linhas de alimentação, o SIG40 substitui o transdutor LIN e o barramento de dados. A sua taxa de transferência triplica a taxa de transferência LIN, permitindo taxas de transferência entre 19,2 e 57,6Kbps. O SIG40 permite ainda reduzir os custos de cada nó e incrementa a capacidade da rede, que funciona numa arquitectura Master/Slave [29, 30].

O SIG40 permite que mais do que uma rede opere sobre a mesma linha, utilizando diferentes frequências da onda portadora, numa gama de 4,5MHz a 6,5Mhz.

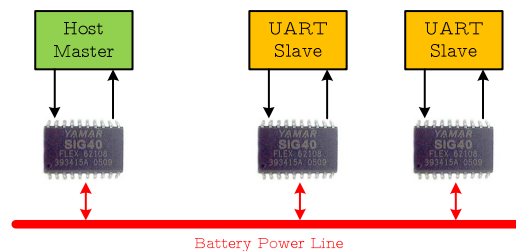


Fig. 21 – Exemplo de uma rede baseada no SIG40

Como principais aplicações do SIG40 estão:

- Electrónica de consumo, como rede interna,
- Barramentos de sensores e actuadores em aviação,
- Sub-barramentos em automóveis para portas, bancos, espelhos, luzes, etc. e,
- Gestão de baterias em carros eléctricos.

Tal como ilustrado na Fig. 22 o SIG40 permite extrair o sinal de dados de uma linha DC com ruído.

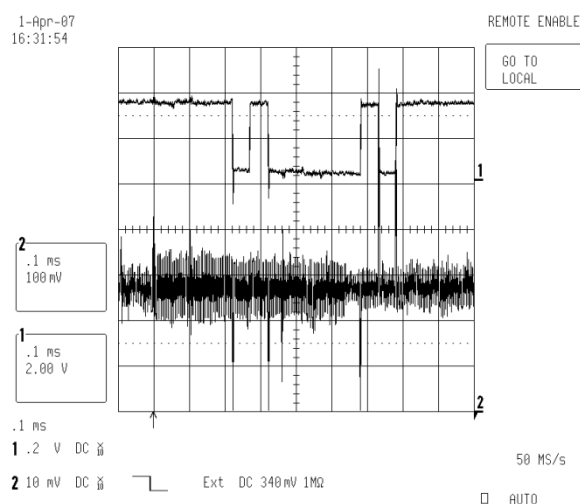


Fig. 22 – Ilustração da extracção do sinal de dados da linha de alimentação, com o SIG40

2.4.2. DCAN250

O DCAN250 é um dispositivo VLSI (Very Large Scale Integration) para comunicações CAN multiplexadas sobre linhas de alimentação. É uma solução completa como substituto da camada física CAN entre uma vasta gama de módulos veiculares como o volante, portas, sensores, ecrãs, Internet/computador de bordo, segurança, painel de controlo, etc.

Este dispositivo actua como um transdutor DC-PLC inteligente para controladores CAN. O DCAN250 contém um modem, um codificador/descodificador de canal, um controlador de comunicação e uma interface como host através de um buffer de mensagens para contornar o ambiente adverso das linhas de alimentação. Quando não existe comunicação no barramento é activado o *Sleep Mode* permitindo assim reduzir o consumo de energia [31].

O DCAN250 permite ainda a utilização de dois canais de comunicação, configurando a frequência de operação entre 2,5MHz ou 5MHz, taxas de transferência de 250Kbps, uma extensão máxima de 12m de cabo DC a uma tensão entre 14 e 42V, até 16 nós em cada canal de comunicação (máximo de 32 nós no total), e ainda mecanismos de correcção de erros [32].

As aplicações principais deste componente são:

- Componentes electrónicos num veículo,
- Controlo de áudio num automóvel,
- Sistemas de segurança e
- Comunicações internas.

2.4.3. DCB500

O DCB500 é um modem transdutor para redes de comunicação multiponto sobre linhas de alimentação DC com ruído. Este dispositivo tem uma interface flexível com o host. Este permite ao utilizador/programador escolher entre comunicação Assíncrona (UART – Universal Asynchronous Receiver/Transmitter) ou Síncrona (SPI – Serial Peripheral Interface), de acordo com a qual preferir.

As mensagens das ECUs são guardadas na sua fila FIFO (First In, First Out) interna. Os dados são então codificados por um código de correcção de erros especialmente desenvolvido para resistir ao ruído típico das linhas de alimentação. Por fim, o sinal é modulado por um modulador QPSK diferencial e é injectado na linha de alimentação através de um filtro passivo [33].

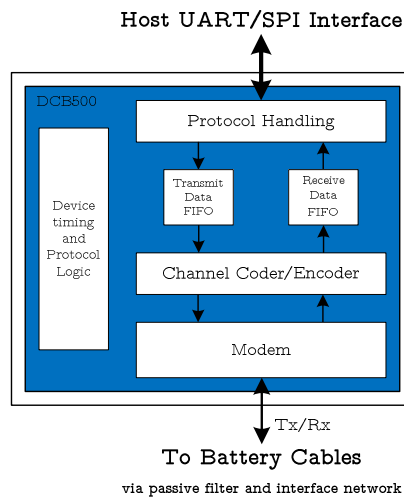


Fig. 23 – Diagrama de blocos de um DCB500

O DCB500 inclui a camada física (Physical Layer) e parte da Link Layer e permite elevadas taxas de transmissão oferecendo a opção de seleccionar entre dois bit rates possíveis, 300 ou 500Kbps. Estas taxas de transmissão são nominais, isto é, em valores médios. A taxa de transmissão no barramento DC é muito superior, tal como indicado em [34].

Tal como indicado no manual do DCB500, este dispositivo está munido de “mecanismos de sincronização, correcção de erros e CSMA/CA. Permite ainda a selecção da frequência de funcionamento do canal, sendo que, na implementação actual, podem ser escolhidas duas frequências de funcionamento, 2,5MHz e 5MHz, estando prevista a implementação de mais duas frequências de funcionamento, 9MHz e 12MHz”.

Um sistema baseado no DCB500 permite a interligação de um máximo de 16 nós por canal e cada nó pode transmitir mensagens de comprimento ilimitado, sobre as linhas DC com ruído, a um dos bit rates possíveis. Do lado do receptor, o bit rate da mensagem recebida é automaticamente detectado [34].

2.4.3.1. Interfaces

Tal como afirmado anteriormente, o DCB500 permite escolher entre duas interfaces com o Host, SPI ou UART.

Na interface SPI são usados cinco pinos para interligação com o Host, HDI (dados do Host para o dispositivo), HDO (dados do dispositivo para o Host), HCLK (sinal de relógio, síncrono, do Host para o dispositivo), HCS (Chip-select do Host para o dispositivo) e HINT (interrupção do dispositivo para o Host). O sinal HINT indica ao Host que o dispositivo tem um novo estado [34].

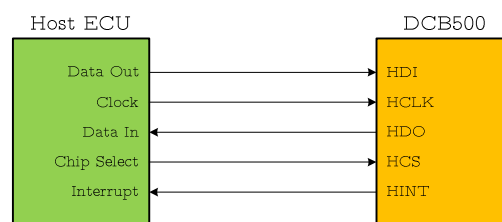


Fig. 24 – Exemplo de ligação SPI entre o DCB500 e o Host

No caso da interface UART existem apenas três pinos. São eles, HDI, HDO e o HCS. Neste modo o sinal HCS é usado para indicar ao dispositivo qual o bit rate do Host.

2.4.3.2. Correcção de erros

O DCB500 possui um mecanismo de correcção de erros (ECC – Error Correction Code) para lidar com erros gerados pelo ruído típico das linhas de alimentação DC. Estão implementados dois tipos de ECC, um para operação a 500Kbps e outro, mais poderoso, para a operação a 300Kbps, tal como indicado em [34].

A cada mensagem transmitida é automaticamente adicionado um checksum. No lado do receptor, a mensagem decodificada é comparada com o checksum e, em caso de erro, é gerada uma interrupção, se esta se encontrar activa.

2.4.3.3. Detecção de colisões

Antes de transmitir para o barramento o dispositivo activa o mecanismo de detecção de colisões. Caso a arbitragem esteja activa, este mecanismo desencadeia o

processo de arbitragem, que dura aproximadamente 100 μ s. Caso a arbitragem esteja desactivada o mecanismo procede ao *sensing* do barramento, durante cerca de 50 μ s, para detectar se existem outras transmissões em curso. Se for detectada actividade no barramento a transmissão é abortada e uma interrupção de colisão é gerada, caso se encontre activa [34].

2.4.3.4. DCB500 Evaluation Board

A Evaluation Board (EVB) do transdutor DCB500 foi concebida para emular e testar a operação do dispositivo num sistema real. Numa mesma linha DC podem comunicar várias destas boards usando os protocolos enunciados anteriormente (UART ou SPI).

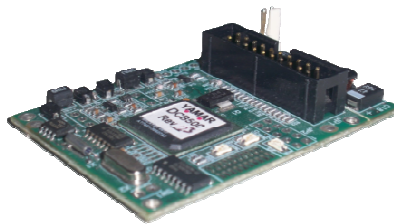


Fig. 25 – Fotografia real de uma EVB DCB500

A EVB contém tudo o que é necessário para o funcionamento do dispositivo, tal como uma rede de protecção da linha, filtros passivos e fonte de alimentação. A board disponibiliza ambos os protocolos e bit rates possíveis. A EVB pode ser conectada directamente ao Host (microcontrolador) usando o conector destinado a essa ligação ou, em alternativa, pode ser conectada a uma porta UART (por exemplo, porta COM¹² de um PC) usando o interface RS232 externo.

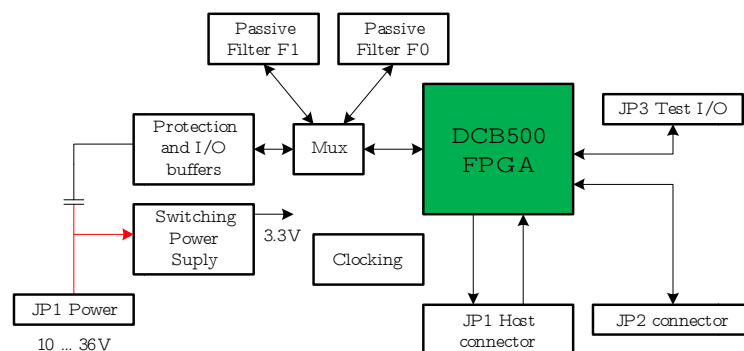


Fig. 26 – Diagrama de blocos da Evaluation Board DCB500

A EVB opera numa gama de alimentação de 10 a 36V DC, com um consumo médio de 80mA a 12V.

O sinal recebido da linha de alimentação passa através de uma rede de protecção, um buffer de entrada e um filtro passivo para um comparador. A FPGA (Field Programmable Gate Array) recebe o sinal digital, processa-o e armazena-o. Os

¹² Porta COM é o nome original, mais frequentemente usado, para a interface de porta série dos PC's.

bytes guardados são então transferidos para o pino HDO do JP1 através de um dos protocolos possíveis UART ou SPI, seleccionado no momento da configuração.

Do lado do transmissor o Host envia os dados para a FPGA através do pino HDI do JP1. Os dados são então modulados e conduzidos através da linha DC.

No entanto, o funcionamento interno do DCB500 é desconhecido, quer em termos de controlo de acesso ao meio quer no que diz respeito à forma como a informação é efectivamente transferida entre nós. Assim, o trabalho aqui apresentado é um exercício de engenharia inversa (reverse engineering) que tem como objectivo descobrir estes aspectos.

3. CARACTERIZAÇÃO TEMPORAL DO CANAL DE COMUNICAÇÃO CAN

Com o intuito de caracterizar o barramento CAN efectuaram-se algumas medidas de tempos de transmissão de mensagens e *throughput*.

Para estas experiências foram utilizados entre dois a quatro nós CAN, baseados no microcontrolador PIC18F258 da Microchip, e um PC para recolha e tratamento dos dados, tal como ilustra a Fig. 27.

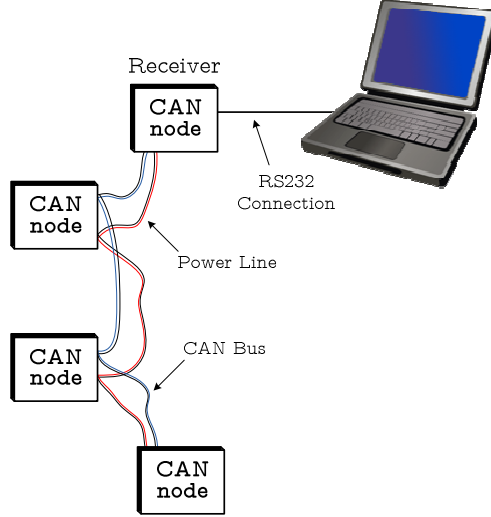


Fig. 27 – Setup experimental das medições sobre o barramento CAN

Cada um dos três nós emissores envia uma mensagem periódica (ID 701, 702 e 703). Como cada nó apenas envia uma mensagem, ao longo deste capítulo usaremos o ID igualmente como relativo ao nó. Em cada nó é configurado um timer que se encontra a correr em free-run e que é utilizado para fazer medidas temporais. Nos nós receptores, a cada mensagem recebida é salvaguardado o valor do timer, como uma marca temporal (timestamp). Os valores do timer são armazenados num buffer de 250 posições, intitulado *buf*, que quando completo é descarregado através da porta série para o PC.

Para o cálculo do tempo entre a chegada de duas mensagens consecutivas o buffer é completamente percorrido, sendo o tempo entre mensagens dado por:

$$buf[i + 1] - buf[i] \quad (1)$$

No caso em que é gerada uma interrupção por overflow do timer, isto é, o timer atingiu o seu valor máximo, 0xFFFF, e passou para 0x0000, a diferença temporal é calculada da seguinte forma:

$$(0xFFFF - buf[i - 1]) + buf[i] \quad (2)$$

Como exemplo, se $buf[i] = 65513 = 0xFFE9$ e $buf[i + 1] = 357 = 0x165$, pela equação (2) o resultado será:

$$(65536 - 65513) + 357 = 380 \text{ ticks}$$

Sabendo que a frequência do *timer* é dada pela frequência de relógio a dividir por quatro, facilmente se obtém o "passo" (ou tick) ou seja o tempo que corresponde a cada unidade de contagem.

$$f_{timer} = \frac{20MHz}{4} = 5MHz \Rightarrow T_{timer} = 200ns$$

Com um pré-scaler de 1:8 obtém-se um *tick* a cada $1.6\mu s$, tal como se observa pelo cálculo seguinte:

$$T_{tick} = \frac{1}{\frac{5MHz}{8}} = 1.6\mu s$$

Sendo o *timer0* do PIC18F258 uma combinação de dois registos de 8bits (TMR0H e TMR0L) tem-se um registo de 16bits, que conta até um máximo de 0xFFFF (65535). Assim, o cálculo do período que decorre até que ocorra *overflow* (passagem do *timer* de 0xFFFF para 0x0000) é imediato:

$$T_{overflow} = 65536 \times 1.6\mu s = 104.9ms$$

Apresenta-se de seguida um exemplo dos resultados esperados no caso em que existem apenas dois nós no barramento, um nó emissor e um nó receptor.

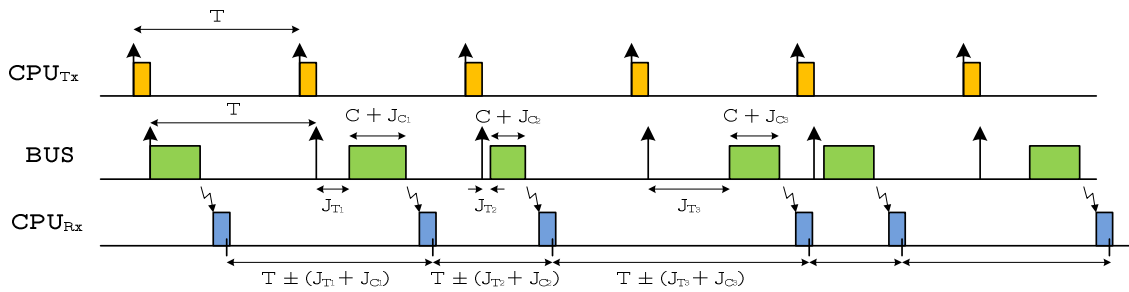


Fig. 28 – Ilustração do *jitter* possível nas mensagens recebidas

O nó emissor (CPU_{Tx}) transmite mensagens periodicamente. Como o acesso ao barramento (BUS) não é imediato existe então um *jitter* (J_T) no período (T) desde que a transmissão é despoletada até que é realmente iniciada. No caso do conteúdo da mensagem ser variável existe ainda um *jitter* (J_C) no seu tempo de execução (C). No nó receptor (CPU_{Rx}), como abordado anteriormente, quando é recebida uma mensagem, é gerada uma interrupção e é lido o valor do *timer0*. Assim, o valor do *timer* lido é afectado pelo menos por estas duas variações.

$$timer0_k = T \pm (J_{T_k} + J_{C_k}), \quad k \in \mathbb{N} \quad (3)$$

Cada uma das experiências efectuadas abordou quatro cenários. O envio de pacotes de dados de tamanho constante e tamanho variável, com e sem sincronismo entre os nós. Excepção é o caso em que existe apenas um nó emissor e um nó receptor. Neste caso foram efectuados envios de pacotes de tamanho constante e tamanho variável, com dois períodos diferentes.

3.1. Um emissor, um receptor

A primeira experiência foi efectuada usando apenas um nó emissor, enviando apenas uma mensagem, ID 701, abordando os quatro cenários descritos para este caso.

3.1.1. ID701, T=5ms, Data Field constante

No primeiro teste é enviada uma mensagem com período de 5ms, com o campo de dados (8 bytes) constante e igual a 0x81...81.

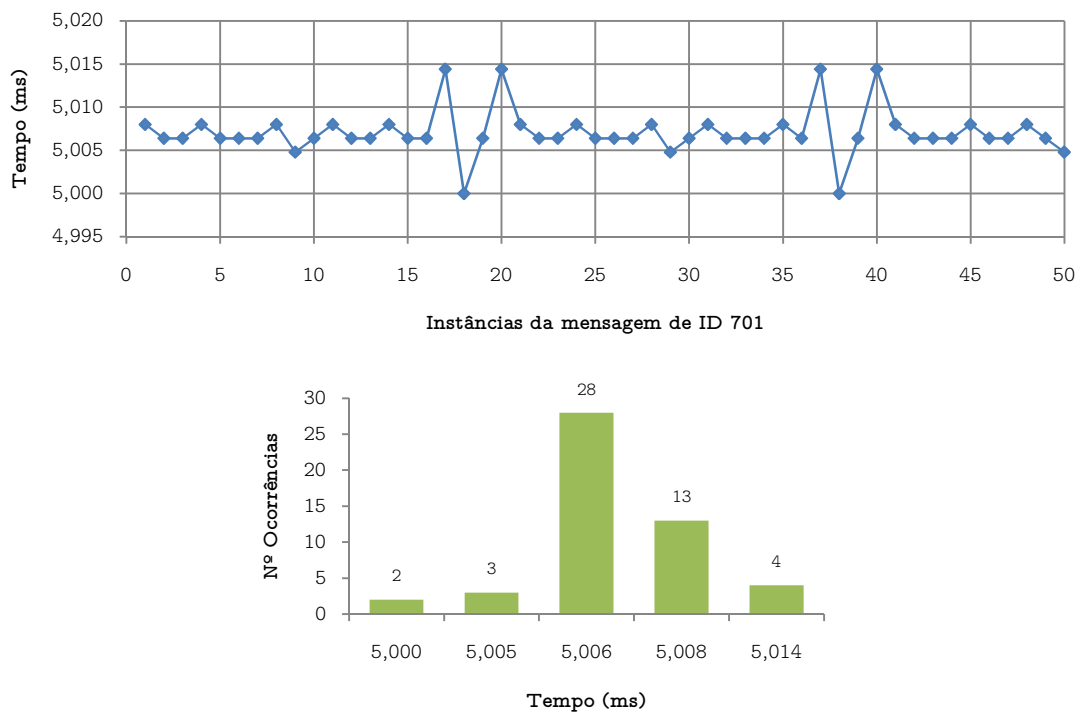


Fig. 29 – ID 701 envia a cada 5ms – DATA = 0x81...81

3.1.2. ID701, T=8ms, Data Field constant

No segundo teste é enviada uma mensagem com período de 8ms, com o campo de dados (8 bytes) também constante e igual a 0x81...81.

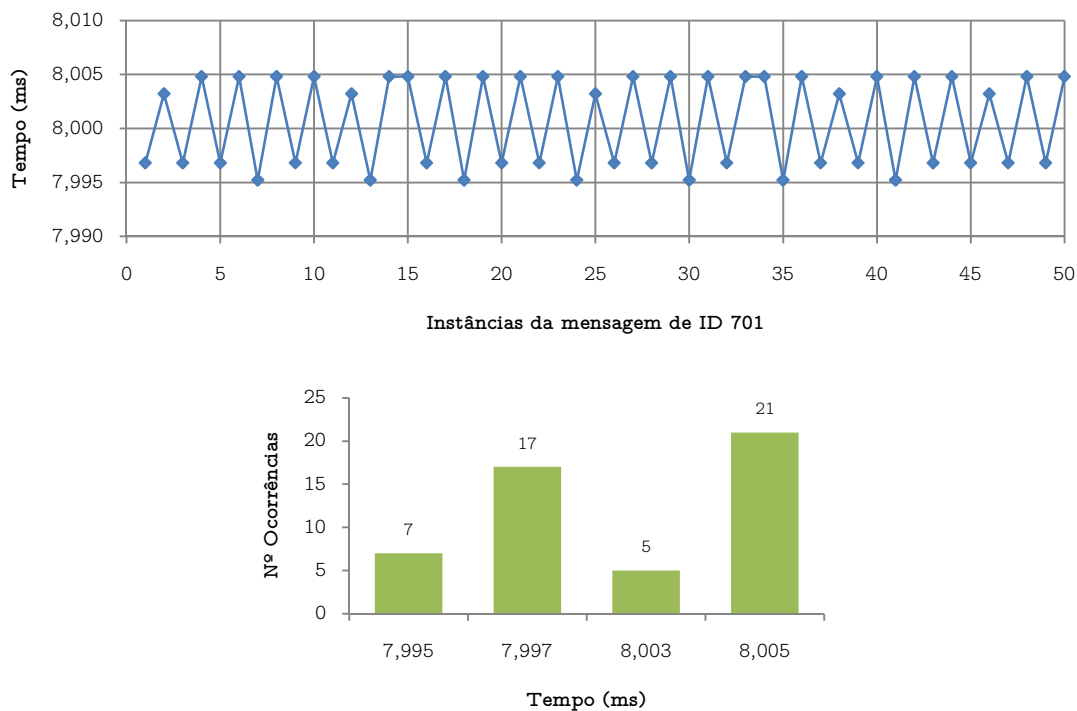
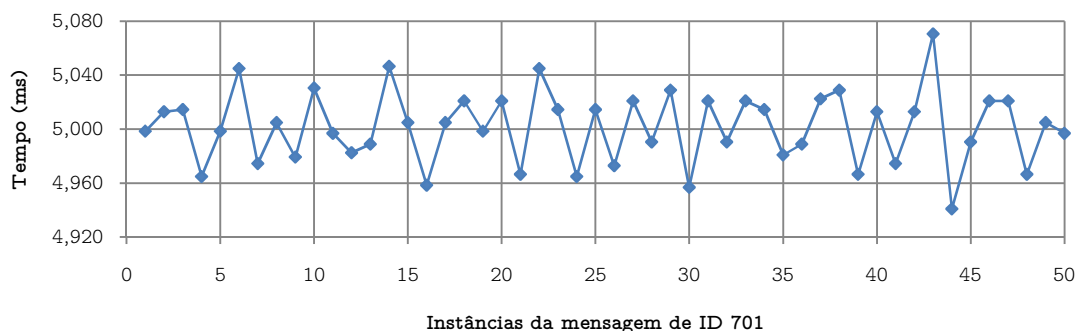


Fig. 30 – ID 701 envia a cada 8ms – DATA = 0x81...81

3.1.3. ID701, T=5ms, Data Field variável

No terceiro teste a mensagem enviada tem período de 5ms e o campo de dados (8 bytes) variável, sendo que, a cada mensagem enviada é incrementado de 0x59.



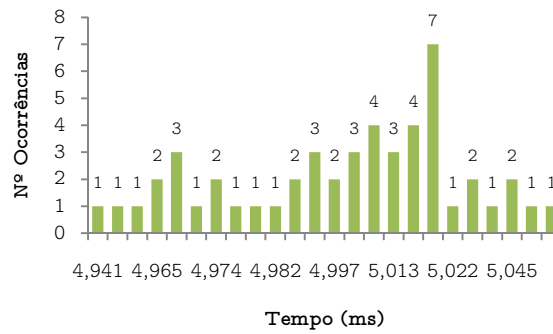


Fig. 31 – ID 701 envia a cada 5ms – DATA = DATA + 0x59

3.1.4. ID701, T=8ms, Data Field variável

No último teste deste caso a mensagem enviada tem período de 8ms e o campo de dados (8 bytes) variável, sendo que, a cada mensagem enviada é incrementado de 0x59.

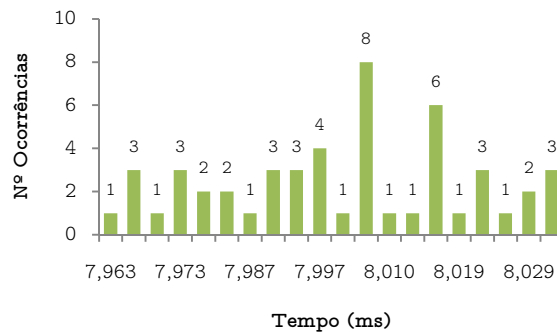
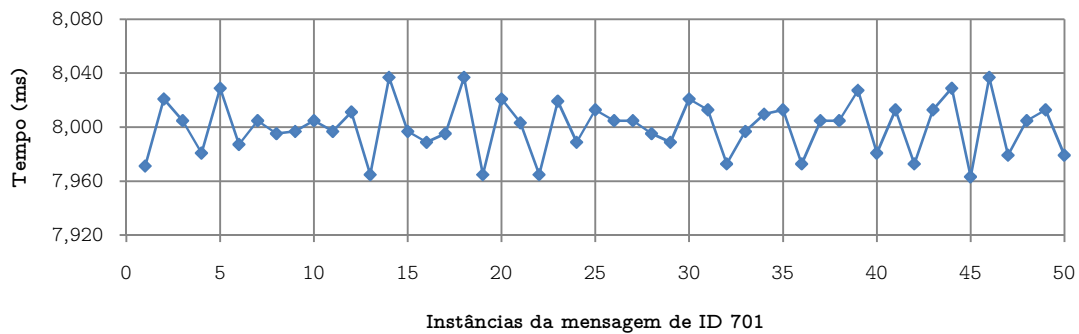


Fig. 32 – ID 701 envia a cada 8ms – DATA = DATA + 0x59

Verifica-se que as mensagens chegam todas com a média esperada, observando-se algumas variações em torno do valor médio.

Para o caso em que o pacote de dados é constante, ver Fig. 29 e Fig. 30, os valores obtidos apresentam pouca variação em torno da média, tal como esperado. Neste caso os valores observados são os apresentados na Tabela 1.

Tabela 1 – Um emissor, um receptor – Data Field constante

Período, T (ms)	Mínimo (ms)	Máximo(ms)	Média (ms)	Desvio Padrão (ms)
5	5,000	5,014	5,007	0,003
8	7,995	8,005	8,000	0,004

Para o caso em que o pacote de dados é variável, ver Fig. 31 e Fig. 32, verifica-se uma grande variação em torno da média. No entanto, este comportamento é o esperado, atendendo à constante variação dos dados enviados e à constante variação do número de stuff bits inseridos no bit stream durante a transmissão. Neste caso os valores observados são os apresentados na Tabela 2.

Tabela 2 – Um emissor, um receptor – Data Field variável

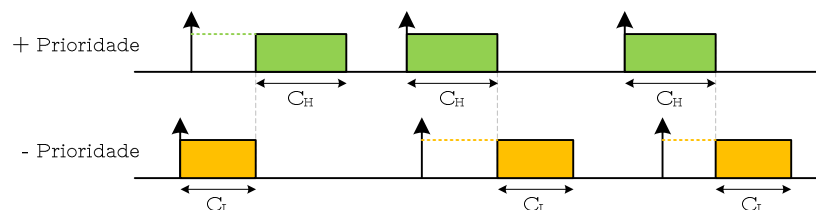
Período, T (ms)	Mínimo (ms)	Máximo(ms)	Média (ms)	Desvio Padrão (ms)
5	4,941	5,070	5,001	0,027
8	7,963	8,037	8,001	0,020

Analisando os valores obtidos, pela regra dos 68-95-99,7¹³, pode afirmar-se que 95% dos valores medidos têm uma variação máxima de $\pm 6\mu s$ e $\pm 8\mu s$ para o caso de pacotes de tamanho constante e período 5 e 8ms, respectivamente.

No caso de pacotes de dados de tamanho variável, 95% dos valores obtidos encontram-se no intervalo $\pm 54\mu s$ para um período de 5ms e $\pm 40\mu s$ para um período de 8ms.

3.2. Dois emissores, um receptor

Neste caso, em que apenas duas mensagens competem pelo barramento, a sua prioridade não é relevante. Visto que não existe preempção, a primeira a ganhar o barramento conclui a sua transmissão sem que seja interrompida, tal como ilustra a Fig. 33.

**Fig. 33** – Duas mensagens no acesso ao barramento

O pior tempo de resposta (WCRT - Worst-Case Response Time) para a mensagem de maior prioridade será o seu tempo de execução (C_H) mais o tempo de bloqueio de não-preempção por parte da mensagem de menor prioridade, ou seja, o tempo de execução da mensagem de menor prioridade (C_L).

¹³ Sendo o desvio padrão representado por σ , a regra dos 68-95-99,7, diz que, numa distribuição normal, 68% dos valores se encontram a $\pm 1\sigma$ do valor médio, 95% dos valores se encontra a $\pm 2\sigma$ e 99,7% dos valores se encontra a $\pm 3\sigma$.

$$WCRT_H = C_H + C_L \quad (4)$$

Para a mensagem de menor prioridade verifica-se exactamente o mesmo.

$$WCRT_L = C_L + C_H \quad (5)$$

3.2.1. ID701 e ID702, T=5ms, Data Field constante

Neste caso, ambos os nós emissores enviam mensagens com uma periodicidade de 5ms, com o campo de dados (8bytes) constante e igual a 0x81...81.

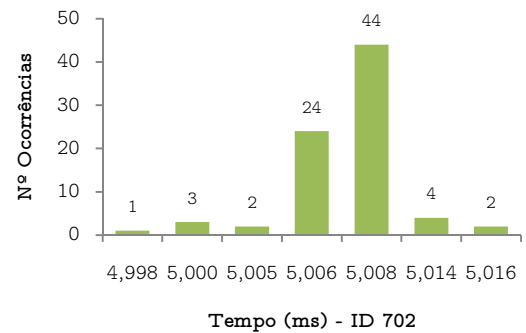
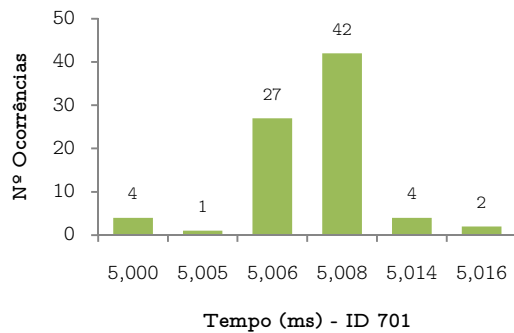
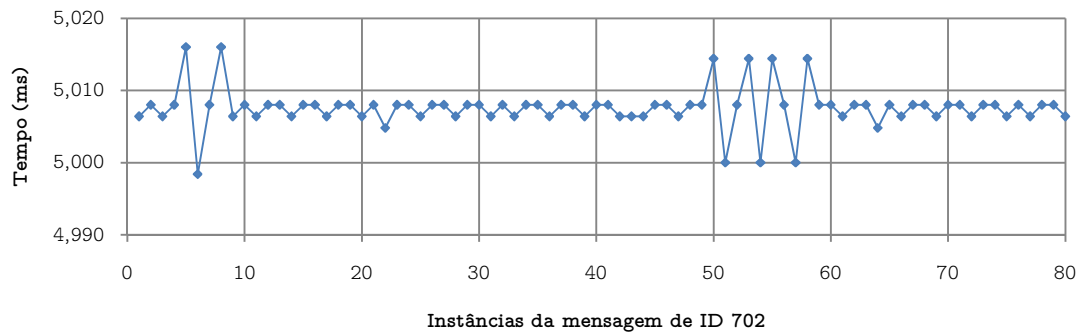
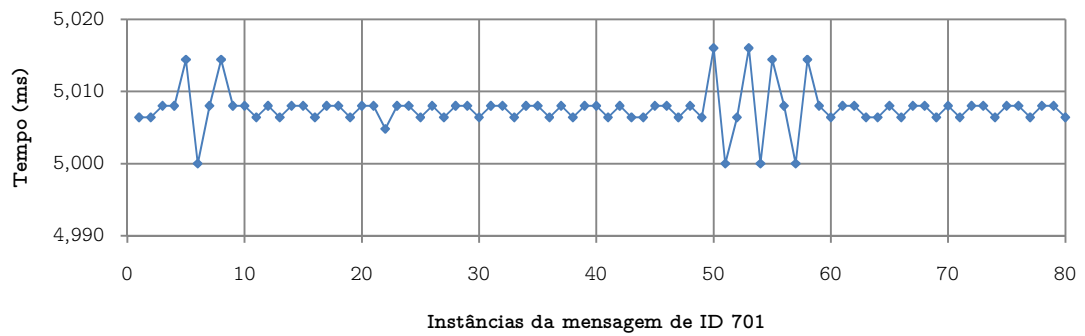


Fig. 34 – ID 701 e ID 702 enviam a cada 5ms – DATA = 0x81...81

3.2.2. ID701, T=8ms e ID702, T=5ms, Data Field constante

Neste caso, o nó com ID701 envia mensagens a cada 8ms e o nó com ID702 a cada 5ms, com o campo de dados (8bytes) constante e igual a 0x81...81.

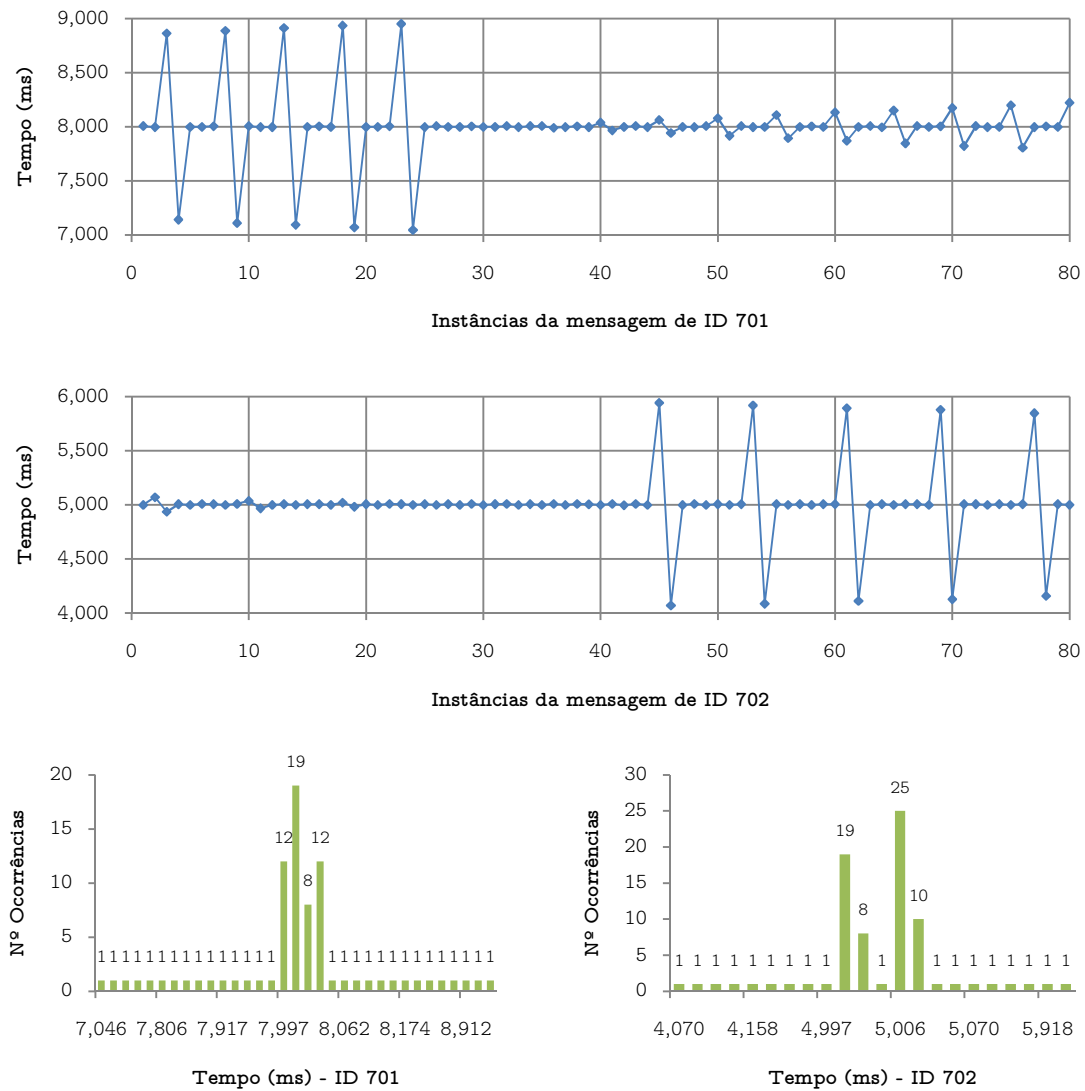


Fig. 35 – ID 701 envia a cada 8ms e ID 702 envia a cada 5ms – DATA = 0x81...81

3.2.3. ID701 e ID702, T=5ms, Data Field variável

Neste caso, ambos os nós enviam mensagens a cada 5ms com o campo de dados (8bytes) variável, incrementando o campo de dados de 0x59 a cada mensagem enviada.

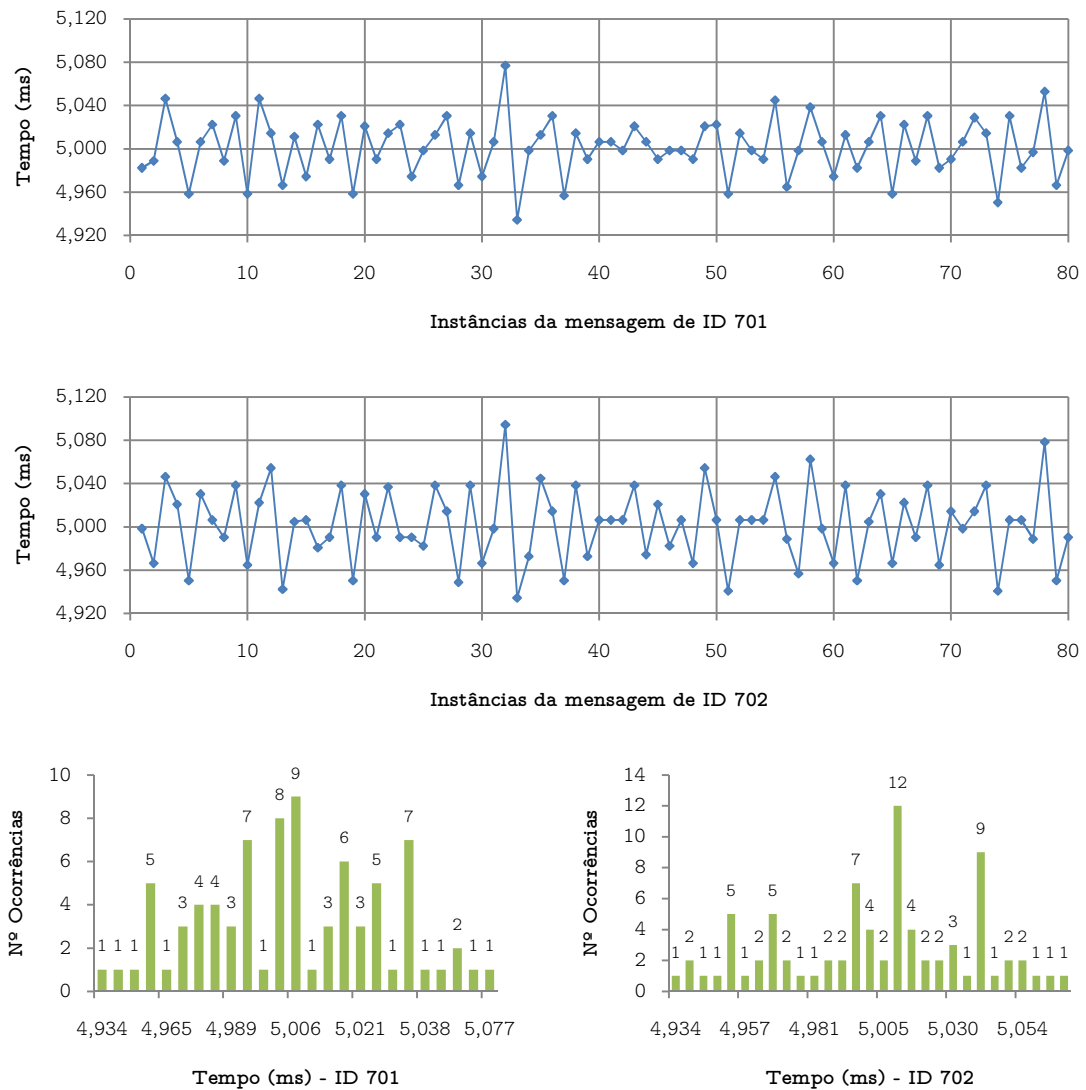


Fig. 36 – ID 701 e ID 702 enviam a cada 5ms – DATA = DATA + 0x59

3.2.4. ID701, T=8ms e ID702, T=5ms, Data Field variável

Neste caso, o nó de ID 701 envia mensagens a cada 8ms e o nó de ID 702 a cada 5ms, incrementando o campo de dados (8bytes) de 0x59 a cada mensagem enviada.

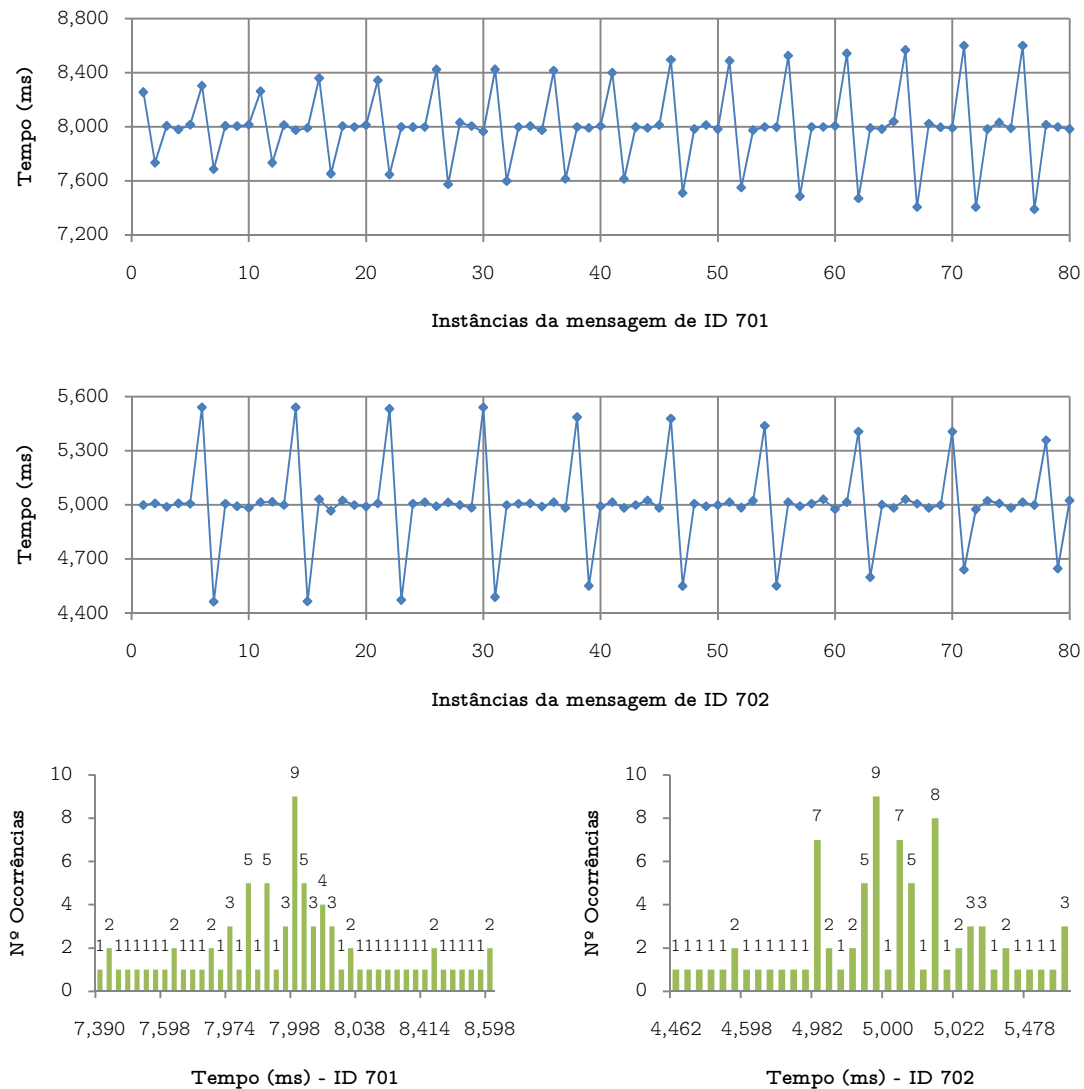


Fig. 37 – ID 701 envia a cada 8ms e ID 702 envia a cada 5ms – DATA = DATA + 0x59

Nos quatro casos aqui apresentados verifica-se que cada mensagem no barramento sofre interferências por parte da outra. Os valores medidos são os apresentados na Tabela 3 e Tabela 4.

Tabela 3 – Dois emissores, um receptor – Data Field constante

ID	Período (ms)	Mínimo (ms)	Máximo (ms)	Média (ms)	Desvio Padrão (ms)
701	5	5,000	5,016	5,008	0,003
702	5	4,998	5,016	5,008	0,003
701	8	7,046	8,950	8,003	0,329
702	5	4,070	5,942	5,003	0,318

No caso em que o pacote de dados é constante e as mensagens têm períodos iguais, ver Fig. 34, os valores obtidos apresentam pouca variação em torno da média, verificando-se que 95% dos valores medidos têm uma variação máxima de $\pm 6\mu s$ (ver Tabela 3).

Para pacotes de dados de tamanho variável e mensagens desfasadas entre si, ver Fig. 35, verifica-se já uma grande variação dos valores medidos em torno da média, verificando-se que 95% dos valores medidos têm uma variação máxima de $\pm 658\mu s$ para a mensagem de maior prioridade e período 5ms e $\pm 636\mu s$ para a mensagem de menor prioridade e período 8ms (ver Tabela 3).

Tabela 4 – Dois emissores, um receptor – Data Field variável

ID	Período (ms)	Mínimo (ms)	Máximo (ms)	Média (ms)	Desvio Padrão (ms)
701	5	4,934	5,077	5,002	0,027
702	5	4,934	5,094	5,002	0,035
701	8	7,390	8,598	8,000	0,286
702	5	4,462	5,541	5,003	0,237

Para o caso em que o pacote de dados é variável e as mensagens possuem o mesmo período, ver Fig. 36, verifica-se que 95% dos valores medidos se encontram a $\pm 54\mu s$ do valor médio para as mensagens de maior prioridade e $\pm 70\mu s$ em torno do valor médio para as mensagens de menor prioridade (ver Tabela 4).

No caso em que o pacote de dados enviado é variável mas as mensagens são enviadas desfasadamente, ver Fig. 37, a variação em torno do valor médio é mais acentuada, verificando-se que 95% dos valores medidos atingem variações de $\pm 572\mu s$ e $\pm 474\mu s$ para as mensagens de maior e menor prioridade, respectivamente (ver Tabela 4).

3.3. Três emissores, um receptor

Neste caso, em que existem 3 nós transmissores, a mensagem de maior prioridade pode ser atrasada, no máximo, do maior dos tempos de transmissão das mensagens de menor prioridade, ou seja,

$$J_{TH_{máx}} = \max\{C_I, C_L\} \quad (6)$$

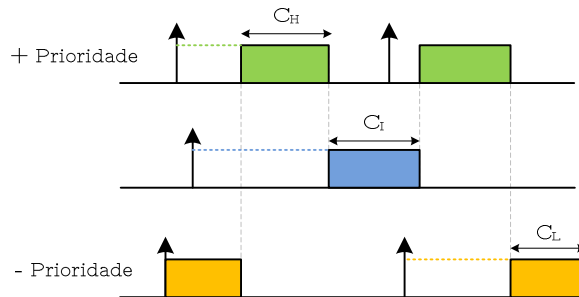


Fig. 38 – Acesso ao barramento – 3 nós transmissores

O pior tempo de resposta ($WCRT$) para a mensagem de maior prioridade será o seu tempo de execução (C_H) mais o maior tempo de bloqueio de não-preempção por parte de uma das mensagens de menor prioridade, C_L ou C_I .

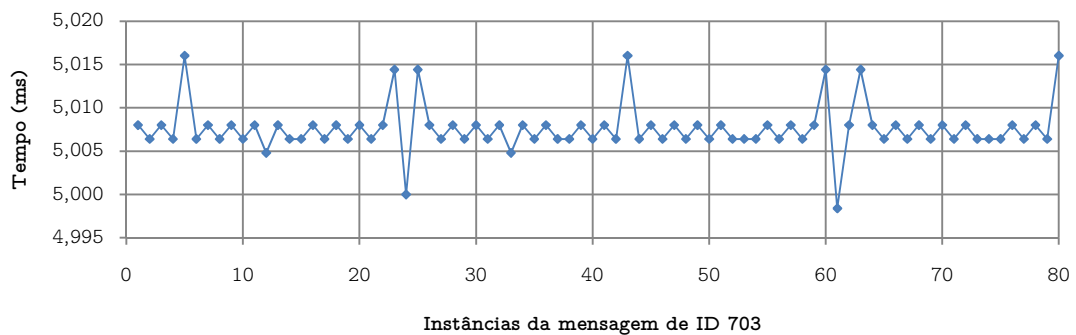
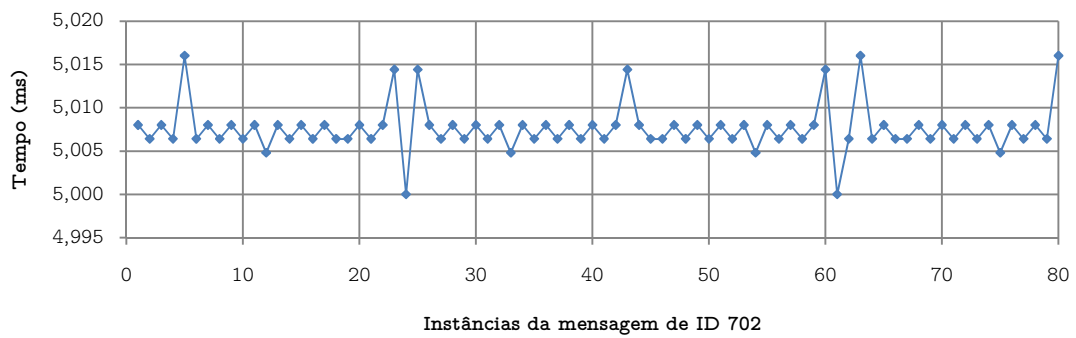
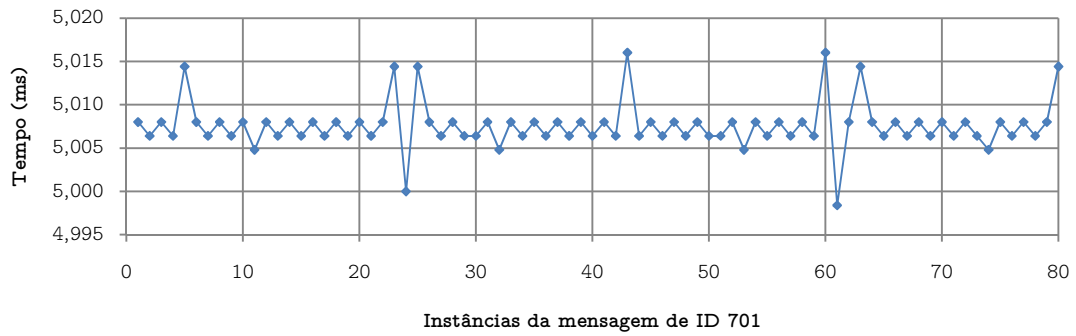
$$WCRT_H = C_H + \max\{C_I, C_L\} \quad (7)$$

As mensagens de menor prioridade podem ser atrasadas, no máximo, da soma dos tempos de transmissão das outras duas mensagens. Assim, para as mensagens de menor prioridade verifica-se que o seu pior tempo de resposta é dado por:

$$WCRT_I = WCRT_L = C_H + C_I + C_L \quad (8)$$

3.3.1. ID701, ID702 e ID 703, T=5ms, Data Field constante

Neste caso, todos os nós emissores enviam mensagens a cada 5ms, com o campo de dados constante e igual a 0x81...81.



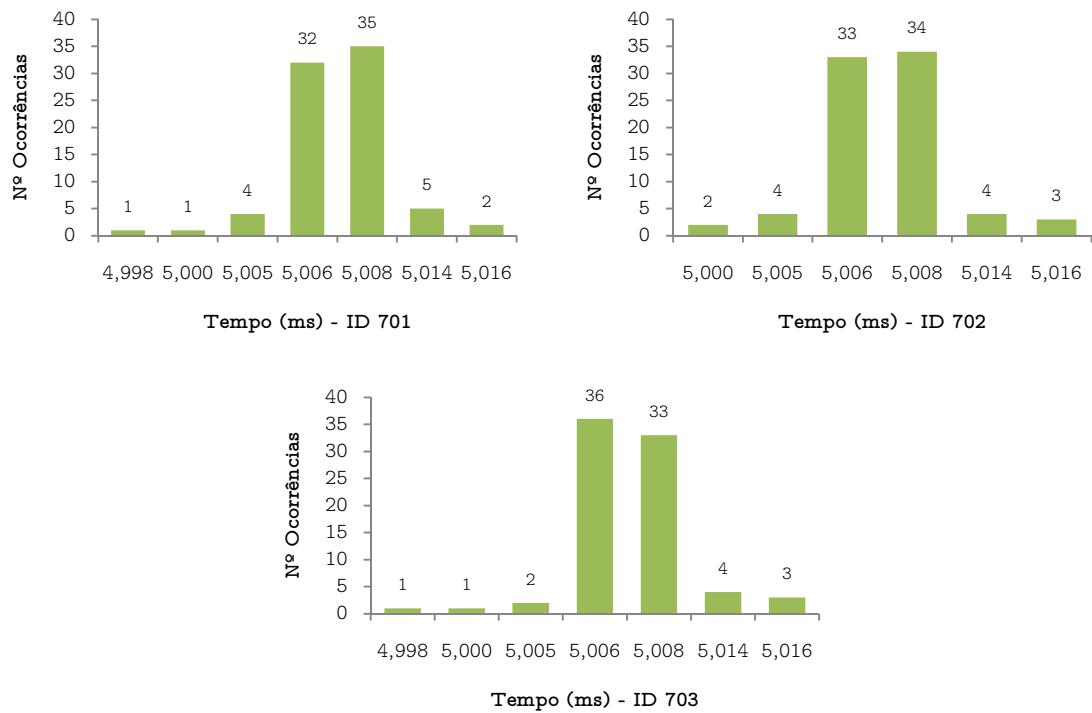
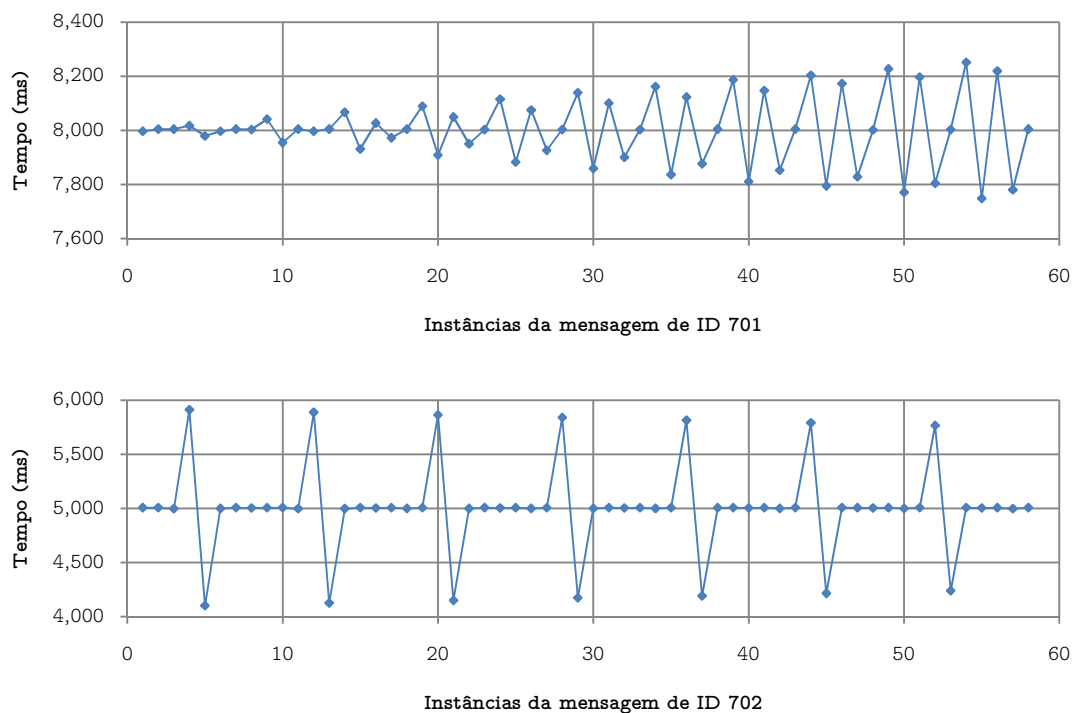


Fig. 39 – ID 701, 702 e 703 enviam a cada 5ms, DATA = 0x81...81

3.3.2. ID701, T=8ms, ID702 e ID 703, T=5ms, Data Field constante

Neste caso, o nó com ID 701 transmite a cada 8ms e os restantes transmitem a cada 5ms, com o campo de dados constante e igual a 0x81...81.



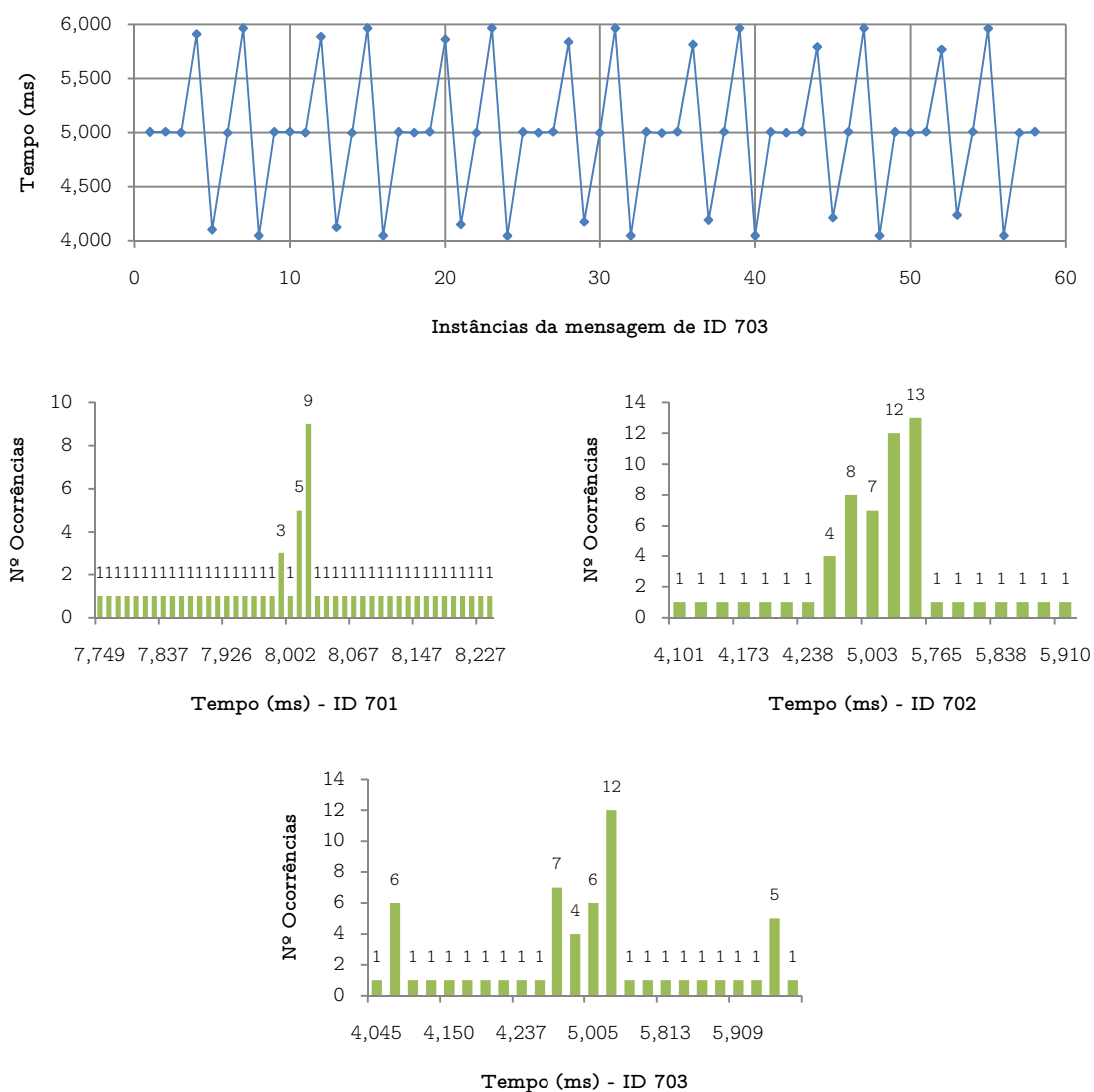
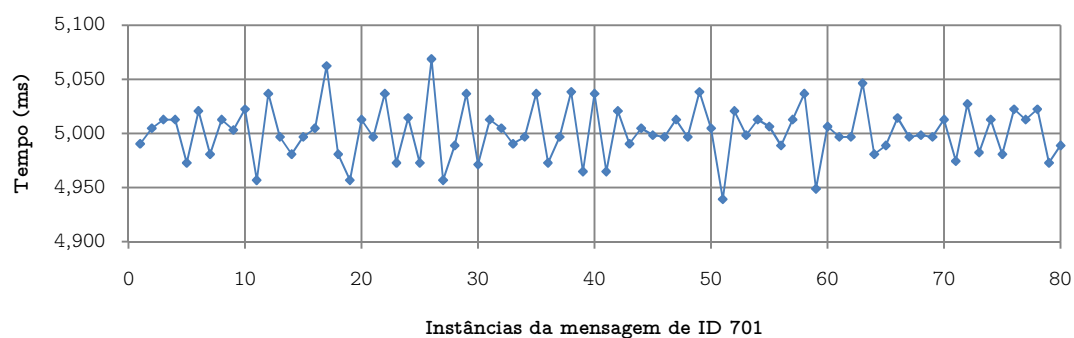


Fig. 40 – ID 701, T=8ms, ID 702 e 703, T=5ms – DATA = 0x81...81

3.3.3. ID701, ID702 e ID 703, T=5ms, Data Field variável

Neste caso, todos os nós emissores enviam mensagens a cada 5ms, com o campo de dados variável, incrementando 0x59 a cada mensagem enviada.



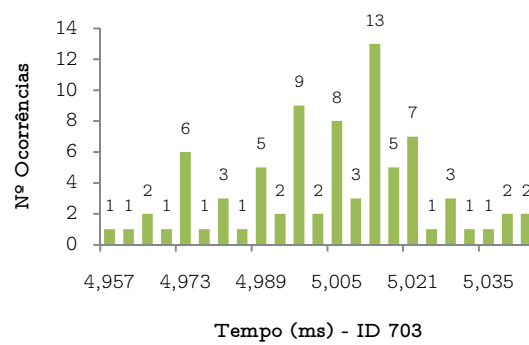
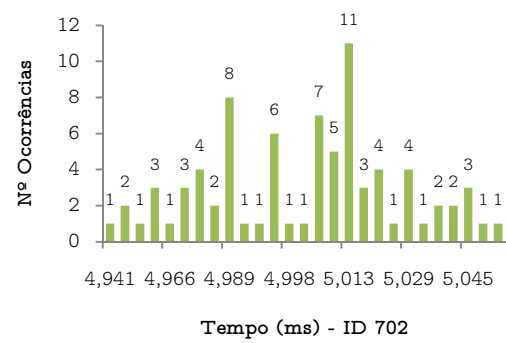
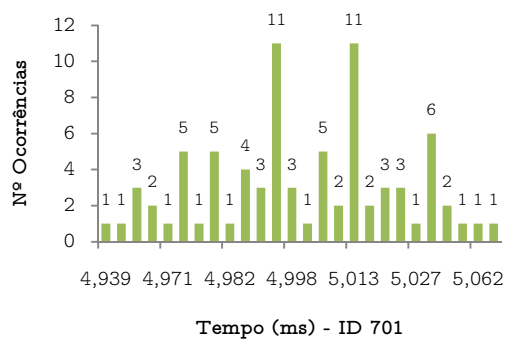
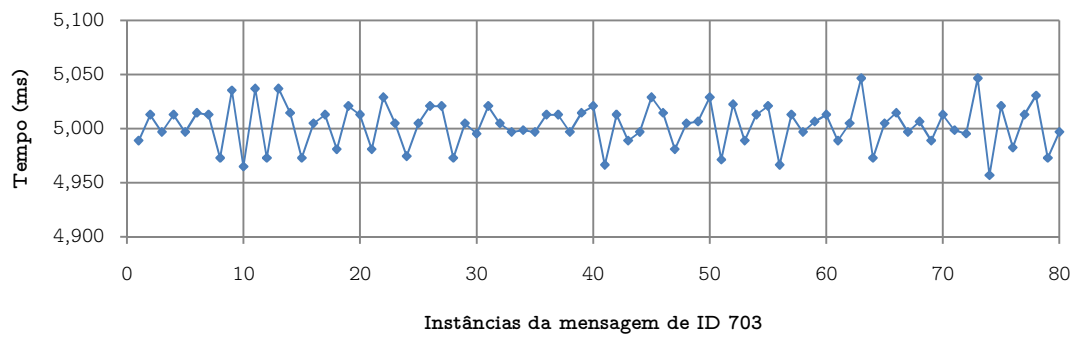
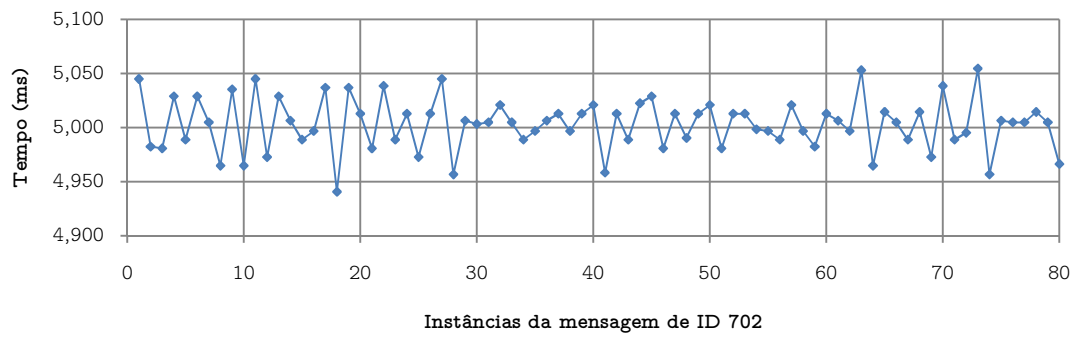
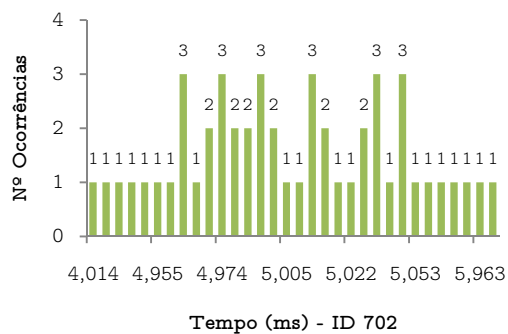
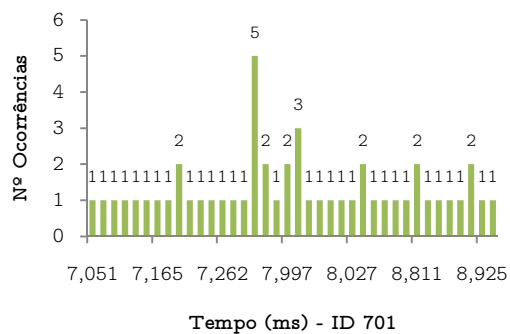
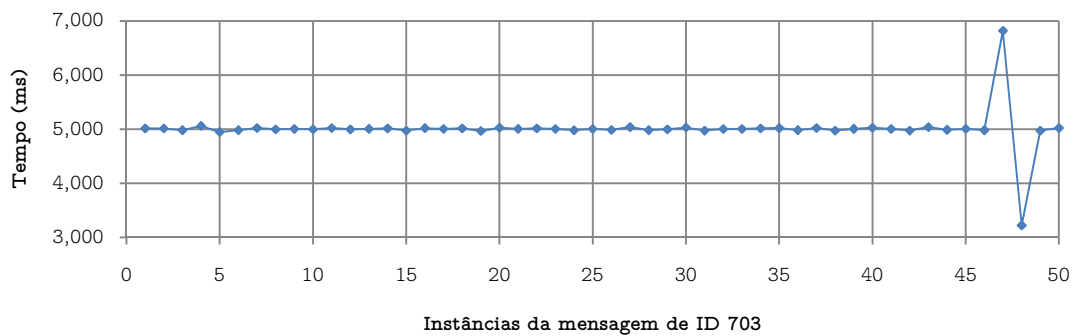
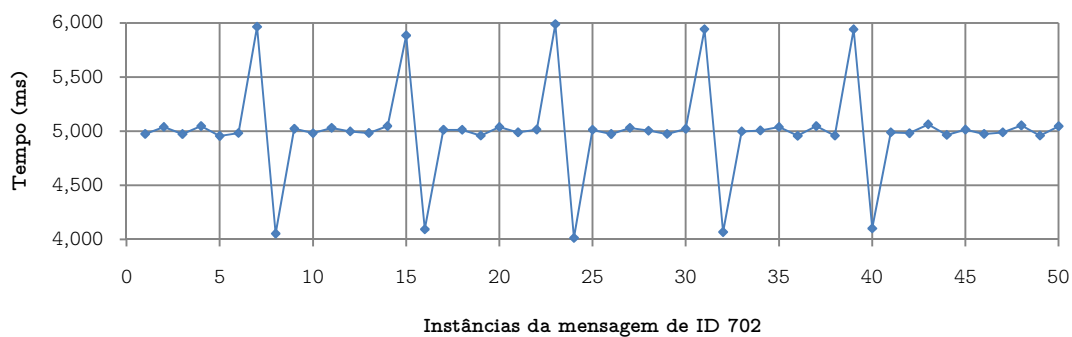
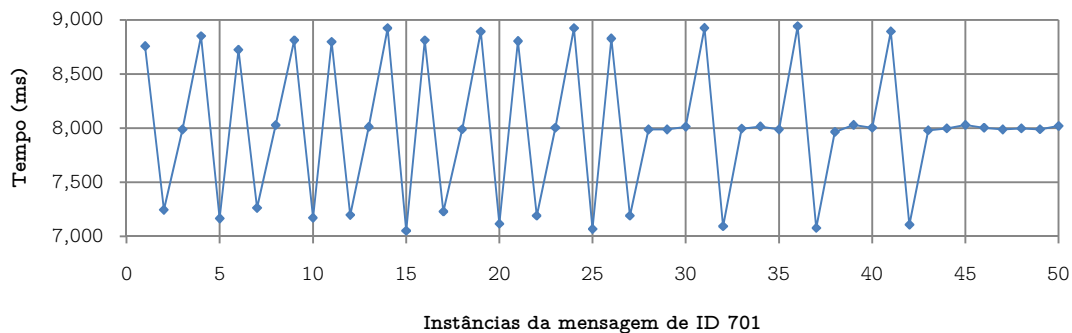


Fig. 41 – ID 701, 702 e 703, T=5ms – DATA = DATA + 0x59

3.3.4. ID701, T=8ms, ID702 e ID 703, T=5ms, Data Field variável

Neste caso, o nó com ID 701 transmite a cada 8ms e os restantes transmitem a cada 5ms, com o campo de dados variável, incrementando 0x59 a cada mensagem enviada.



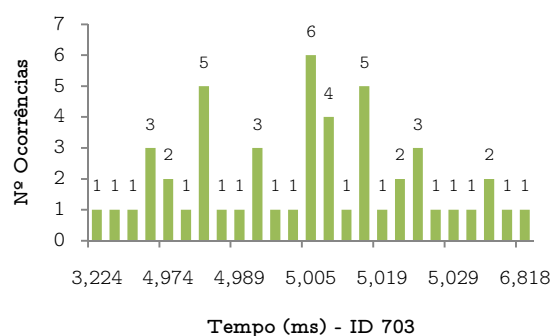


Fig. 42 – ID 701, T=8ms e 702 e 703, T=5ms – DATA = DATA + 0x59

Nos quatro casos apresentados nas secções anteriores, verifica-se que cada mensagem no barramento sofre interferências por parte das restantes. Os valores medidos são os apresentados na Tabela 5 e Tabela 6.

Tabela 5 – Três emissores, um receptor – Data Field constante

ID	Período (ms)	Mínimo (ms)	Máximo (ms)	Média (ms)	Desvio Padrão (ms)
701	5	4,998	5,016	5,008	0,003
702	5	5,000	5,016	5,008	0,003
703	5	4,998	5,016	5,008	0,003
701	8	7,749	8,251	8,001	0,124
702	5	4,101	5,910	5,003	0,414
703	5	4,045	5,965	5,003	0,630

Verifica-se que existe uma variação mínima de 95% dos valores medidos em torno da média obtida, de $\pm 6\mu s$ para o caso em que o campo de dados enviado é constante e as mensagens possuem o mesmo período, igual a 5ms, ver Fig. 39.

No caso em que é enviado um pacote de dados constante e o nó de ID 701 está desfasado dos restantes, ver Fig. 40, os valores obtidos são bastante diversos. No entanto, os valores obtidos são coerentes com o resultado expectável. Tal como ilustrado no início da secção 3.3 verifica-se que a mensagem de maior prioridade é a que sofre menor variação no seu tempo de chegada ao receptor, visto que apenas pode ser afectada pelo máximo dos tempos de execução das mensagens de menor prioridade. No caso do nó de prioridade intermédia verifica-se que 95% das suas mensagens têm um tempo de entrega ao receptor de $\pm 828\mu s$ e que para o nó de menor prioridade a variação máxima em torno da média é de $\pm 1260\mu s$. Estes valores explicam-se pelo facto de as mensagens destes nós poderem sofrer interferência um do outro e ainda da mensagem de maior prioridade.

No cenário em que existem três mensagens a competir pelo barramento e o envio é sincronizado (T=5ms) e o pacote de dados enviado é variável (Fig. 41) verificam-se variações em torno do valor médio entre $\pm 40\mu s$ e $\pm 50\mu s$, para 95% dos valores medidos (Tabela 6).

Por outro lado, no caso em que a mensagem de maior prioridade se encontra desfasada das restantes e o pacote de dados é variável (Fig. 42), 95% dos valores

obtidos variam entre $\pm 728\mu\text{s}$ e $\pm 1284\mu\text{s}$ em torno dos respectivos valores médios medidos (Tabela 6).

Tabela 6 – Três emissores, um receptor – Data Field variável

ID	Período (ms)	Mínimo (ms)	Máximo (ms)	Média (ms)	Desvio Padrão (ms)
701	5	4,939	5,069	5,001	0,025
702	5	4,941	5,054	5,003	0,024
703	5	4,957	5,046	5,003	0,020
701	8	7,051	8,941	8,001	0,642
702	5	4,014	5,987	5,003	0,425
703	5	3,224	6,818	5,004	0,364

4. CARACTERIZAÇÃO TEMPORAL DO CANAL DE COMUNICAÇÃO POWERLINE

O canal de comunicação Powerline foi construído com base nas Evaluation Boards (EVB) do modem transdutor DCB500 da empresa Israelita Yamar Electronics e nas placas FlexBoard (Full Version) da empresa Italiana, Evidence¹⁴, usadas como Host. As FlexBoard (Full Version) são concebidas para exportarem todas as ligações de um controlador dsPIC® DSC padrão. No caso presente o controlador utilizado foi o dsPIC33FJ256MC710.

Tal como apresentado na secção 2.4.3., o modem DCB500 permite escolher entre dois tipos de ligação com o Host, UART ou SPI. Para o desenvolvimento deste trabalho foi eleita a ligação SPI. Esta escolha deveu-se ao facto da ligação SPI permitir usufruir de uma variedade de características da EVB muito superior aquela que seria possível com uma ligação UART. A ligação SPI permite assim acesso a registos e o uso de interrupções que o modo UART não permite, tais como registo de interrupções, configuração do dispositivo e de comunicação, arbitragem, etc. Todos os registos são lidos e escritos na forma *LSB first*, isto é, a leitura/escrita iniciam-se pelo bit menos significativo. Acontecendo o mesmo na transmissão se dados, tal como indicado no manual do modem transdutor [34].

O DCB500 é controlado por um conjunto de comandos de controlo e construção de mensagens. A Tabela 7 apresenta os comandos disponíveis no modo SPI apresentados no manual do modem transdutor DCB500.

Tabela 7 – Comandos de controlo do modem transdutor DCB500

Comando	Descrição
Tx_MSG3	Inicia a transferência de uma mensagem a 300Kbps. O cabeçalho é seguido do código de correcção de erros e de seguida são enviados os bytes de dados.
Tx_MSG5	Inicia a transferência de uma mensagem a 500Kbps. O cabeçalho é seguido do código de correcção de erros e de seguida são enviados os bytes de dados.
Frame_End	Termina a transmissão de uma mensagem.
Rx_MSG	O Host lê o buffer de recepção (Rx buffer) do dispositivo (DCB500). Quando o último byte do buffer é lido o dispositivo gera uma interrupção para o Host.
Write_REG_X	Escreve para o registo X do dispositivo.
Read_Interrupt	Após um sinal de interrupção gerado pelo dispositivo o Host lê o registo de interrupções do dispositivo, que contém o código da interrupção gerada.
Read_REG_X	Lê o registo interno X do dispositivo.
CLEAR_Rx_MSG	Limpa o restante de uma mensagem que se encontre na fila FIFO de recepção
CLEAR_Tx_FIFO	Limpa a fila FIFO de transmissão
NOP	Não efectua qualquer operação

¹⁴ <http://www.evidence.eu.com/>

Assim, para transmitir uma mensagem para o barramento DC, segundo a descrição dada pela Yamar, um de dois comandos tem que ser escrito, Tx_MSG3 ou Tx_MSG5, para envio da mensagem a 300 ou 500Kbps, respectivamente. Se o dispositivo não detectar outra transmissão no barramento os bytes de dados são escritos para a fila FIFO do dispositivo (DCB500), obtém o controlo do barramento e inicia a transmissão dos bytes de dados. Este abandona o controlo do barramento apenas depois de ter sido enviado o comando Frame_End.

As filas FIFO de transmissão e de recepção, consistem em 32 bytes cada uma permitindo assim transferências simples de mensagens com menos de 32 bytes (Single Message).

O manual descreve ainda a possibilidade de envio de fluxos de dados (Stream Message) monitorizando as interrupções Tx_FIFO_Full e Tx_FIFO_1/4. Quando a fila FIFO de transmissão está cheia o dispositivo gera uma interrupção Tx_FIFO_Full, caso esteja activa, para o Host. Nesse caso, o Host tem que parar a transferências de dados, gerir a interrupção e esperar que ocorra a interrupção Tx_FIFO_1/4 para indicar que podem ser transferidos novos dados para o dispositivo.

Do lado do receptor, quando este detecta uma transmissão em curso, o dispositivo sinaliza o Host com uma interrupção NEW_MSG, se estiver activa. O Host pode então ler a mensagem enviando para isso o comando Rx_MSG, o qual inicia a transferência dos dados do dispositivo para o Host. Quando não existirem mais dados na fila FIFO é gerada a interrupção Rx_FIFO_Empty, enquanto o Host está a ler o último byte, se estiver activa.

Após cada leitura do registo de interrupções o dispositivo faz o seu reset, isto é, coloca o registo de interrupções a 0x00 – No Interrupt.

Na Tabela 8 estão representadas as interrupções disponíveis no modem DCB500, tal como ilustrado no seu manual.

Tabela 8 – Descrição das interrupções do modem DCB500

Interrupção	Descrição
No Interrupt	Não existe qualquer interrupção. O registo foi reiniciado, 0x00.
Tx FIFO Empty	A fila FIFO de transmissão foi esvaziada.
Tx FIFO 1/4	A fila FIFO de transmissão decresceu para $\frac{1}{4}$ do seu tamanho.
Tx FIFO Full	A fila FIFO de transmissão está cheia.
WakeUp	O dispositivo saiu do modo adormecido – Sleep Mode.
Collision	Foi detectada uma transmissão em curso ou o dispositivo perdeu o processo de arbitragem
Not Valid	O último comando escrito não é válido.
Rx FIFO Empty	Não existem mais bytes para serem lidos da fila FIFO de recepção.
Rx FIFO 3/4	O buffer de recepção atingiu $\frac{3}{4}$ da sua capacidade.
Rx FIFO Overflow	O buffer de recepção ultrapassou a sua capacidade de armazenamento. É considerado um erro de recepção.
Error Detect	Foi detectado um byte corrompido na mensagem recebida. Quando é detectado um erro, a recepção pára.

New Message	Uma nova mensagem chegou e está pronta a ser lida no buffer de recepção.
Message End	Fim de mensagem recebido. Código de interrupção: 1xxxxxxb ¹⁵ . Restam xxxxxxxb bytes no buffer de recepção, para serem lidos.

A Fig. 43 ilustra a montagem experimental efectuada – a ligação SPI DCB500/Host, o barramento DC Powerline, fonte de alimentação (bateria), modem transdutor (DCB500) e o Host (FlexBoard).

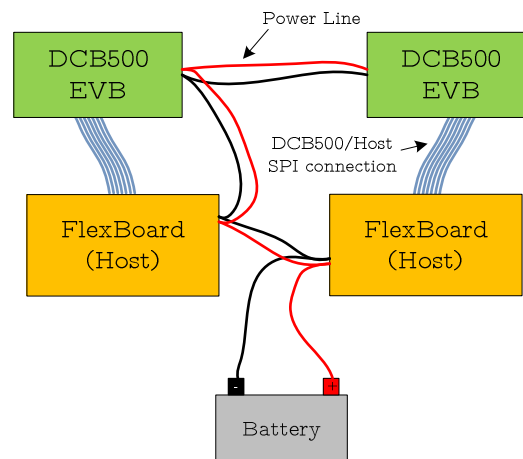


Fig. 43 – Montagem experimental do canal de comunicação Powerline

4.1. Tempo de transmissão de uma mensagem

O primeiro teste efectuado sobre o canal de comunicação PLC foi o de medição do tempo de transmissão de uma mensagem através do canal apresentado na secção anterior.

Tal como ilustrado na Fig. 43, existem dois nós, um emissor e outro receptor. Nesta experiência foi ainda adicionada uma ligação que interliga os Hosts (FlexBoards) através de um pino, que tem como finalidade sinalizar o emissor quando o receptor receber uma mensagem.

Assim, para a medição destes valores foi desenvolvido e utilizado o algoritmo apresentado na Fig. 44.

¹⁵ O sufixo *b* indica o formato binário. Existem várias notações possíveis para o formato binário, como por exemplo, 1010 binário = 1010b = 1010B = 1010₂ = 0b1010, sendo todas elas equivalentes.

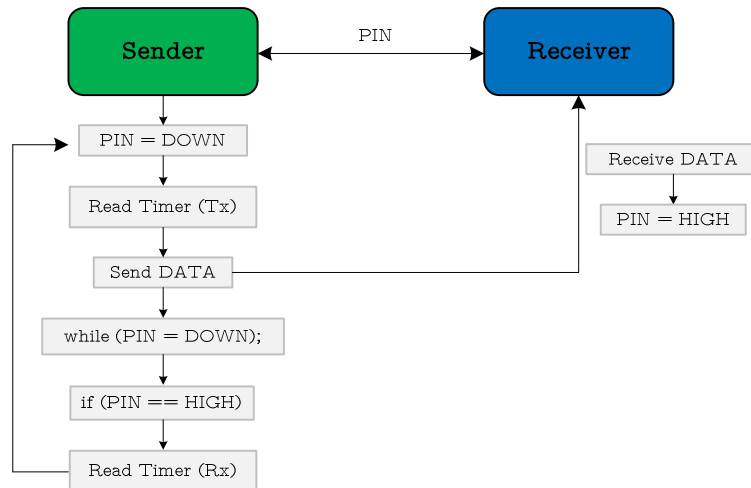


Fig. 44 – Algoritmo para medida do tempo de transmissão de uma mensagem

Do lado do emissor começa-se por colocar o pino de interligação a zero (DOWN) – mensagem não recebida, é lido o valor do *timer*, é enviada a mensagem e enquanto o pino estiver a zero, espera pela sinalização do receptor.

O receptor tem como única função receber a mensagem e colocar o pino de interligação a alto (HIGH) – mensagem recebida.

É capturado o valor do *timer* imediatamente antes de enviar a mensagem e novamente quando é detectado o PIN a 1, tal como se observa na Fig. 44. De seguida é efectuada a subtracção destes dois valores, obtendo-se o tempo de transmissão da mensagem. O processo de medida é ilustrado na Fig. 45.

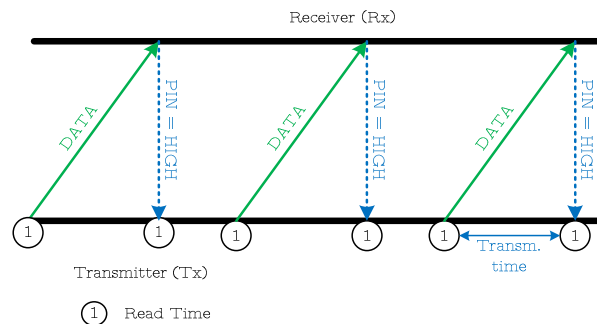


Fig. 45 – Ilustração do processo de medida

Foram efectuadas medidas do tempo de transmissão de mensagens de tamanho compreendido entre 1 e 4096 bytes. Os resultados obtidos são os apresentados nas figuras seguintes. De notar a escala horizontal logarítmica dos gráficos de mensagens de tamanho compreendido entre 64 e 4KB.

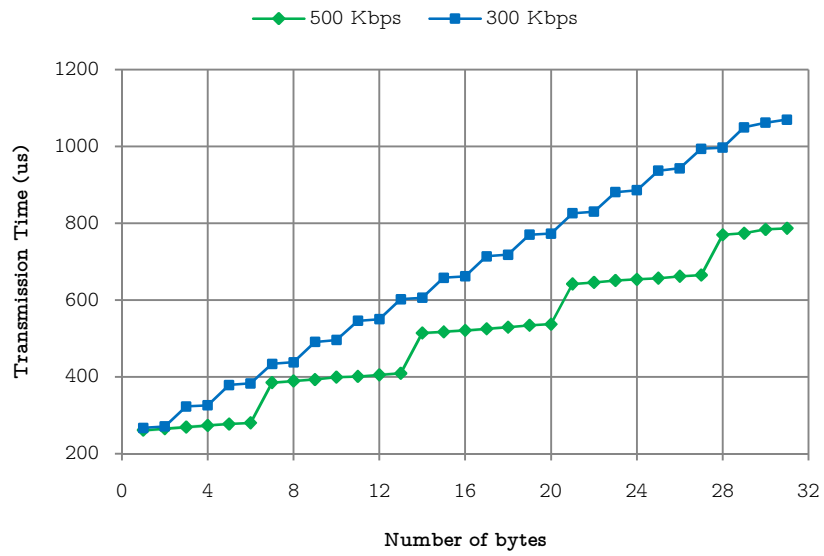


Fig. 46 – Tempo de transmissão – 1 a 31 bytes – Single Message Mode

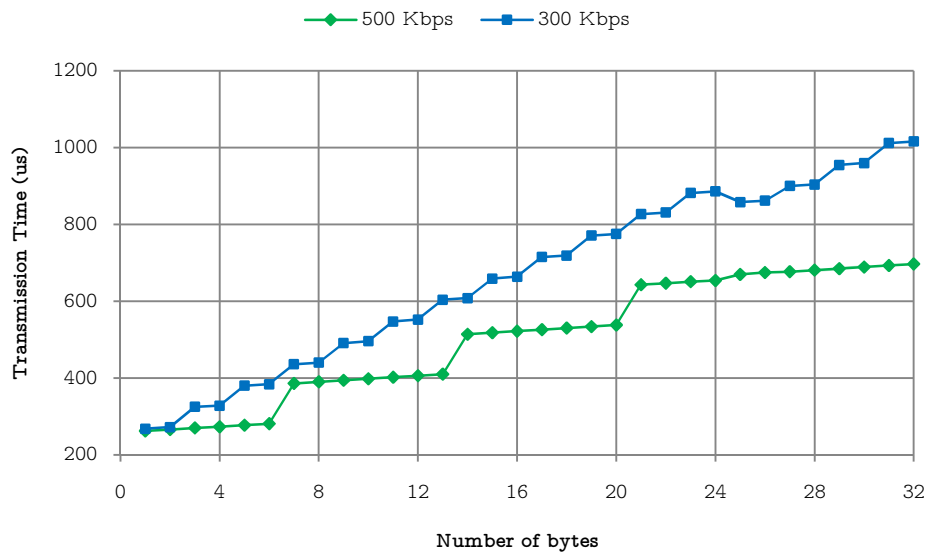


Fig. 47 – Tempo de transmissão – 1 a 32 bytes – Stream Mode

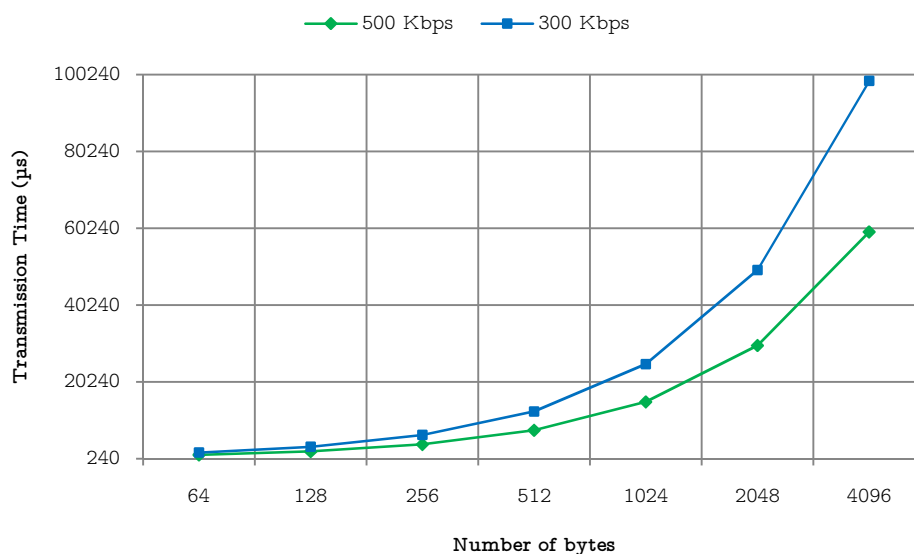


Fig. 48 – Tempo de transmissão – 64 e 4096 bytes – Stream Mode

Observando os valores obtidos para pacotes de dados de tamanho compreendido entre 1 e 4KB¹⁶, Fig. 46, Fig. 47 e Fig. 48, verifica-se que o tempo de transmissão aumenta com o aumento do tamanho do pacote de dados, tal como esperado. Adicionalmente, confirma-se que o processo de transmissão é efectuado de igual forma, no modo *Stream Message* e no modo *Single Message*, para pacotes de tamanho igual ou inferior a 24 bytes.

Verifica-se ainda que, para 300Kbps, usar *Stream mode* ajuda a diminuir o tempo de transmissão medido.

Concluindo, observa-se claramente que as mensagens são fragmentadas e enviadas em pequenos pacotes elementares, ou seja, por cada conjunto de bytes enviados é enviado um pacote com o respectivo *overhead*. Para 300Kbps o tamanho do pacote elementar é de 2 bytes ao passo que para 500Kbps verifica-se que o tamanho do pacote elementar é de 7 bytes, à excepção do primeiro pacote que é de apenas 6 bytes.

Com base nos valores de tempo medidos foi possível calcular o bit rate efectivo da transmissão para todos os casos apresentados, ver Fig. 49, Fig. 50 e Fig. 51.

¹⁶ KB – KiloByte, é uma unidade de armazenamento de informação digital. 1KB = 1024 bytes = 8192 bits

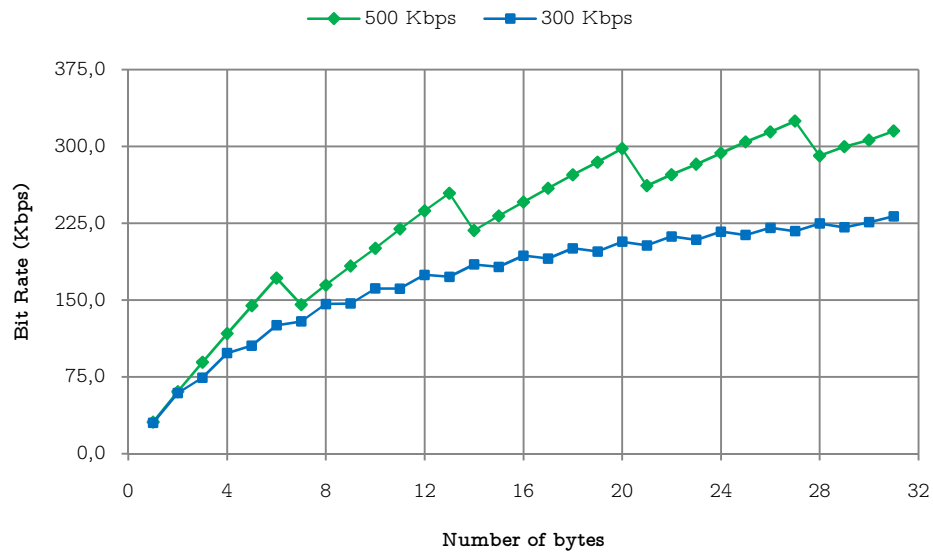


Fig. 49 – Bit Rate para mensagens entre 1 e 31 bytes – Single Message Mode

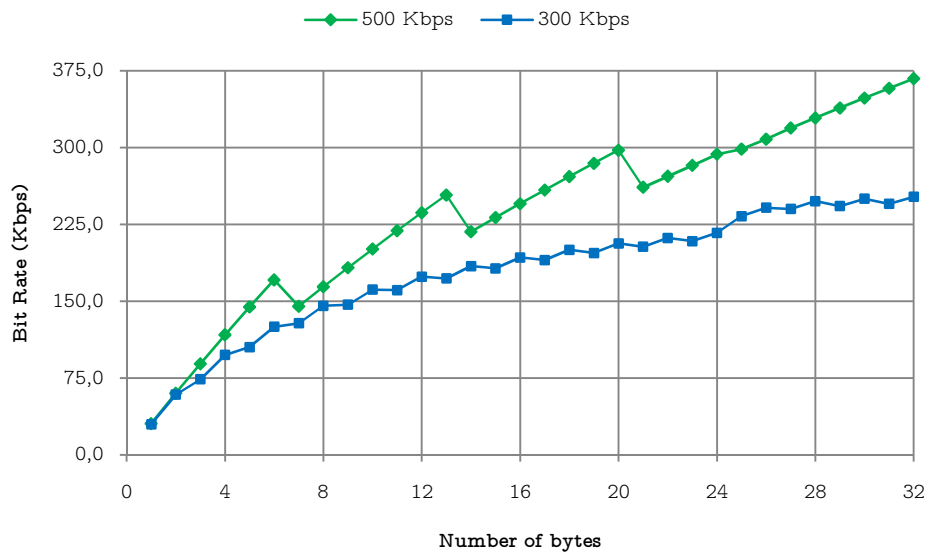


Fig. 50 – Bit Rate para mensagens entre 1 e 32 bytes – Stream Mode

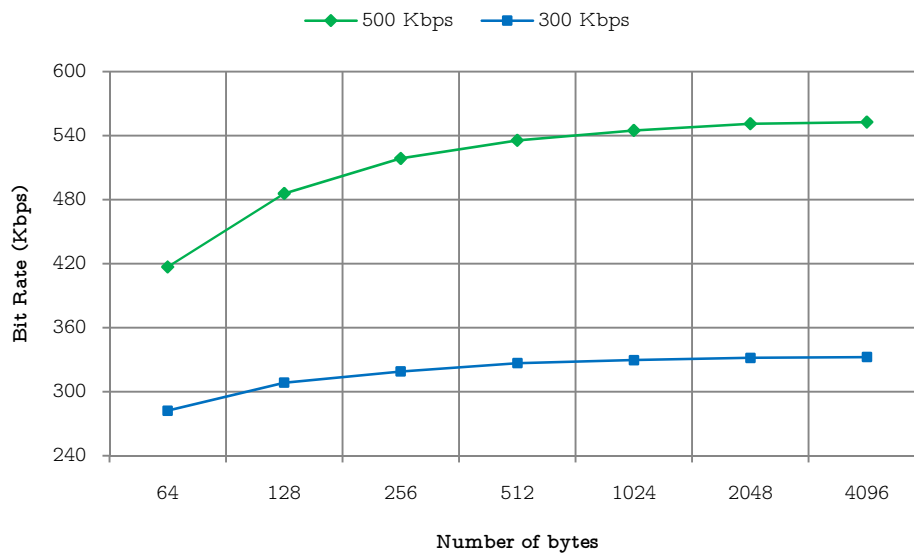


Fig. 51 – Bit Rate para mensagens entre 64 e 4096 bytes – Stream Mode

Como se observa para ambos os modos (300 e 500Kbps) os seus valores nominais são atingidos e ultrapassados mas apenas para o *Stream Message Mode* e transmissões de pelo menos 100 e 170 bytes, para 300 e 500Kbps, respectivamente. Para transferências de pacotes de tamanho inferior a 24 bytes verifica-se que os valores atingidos são sempre inferiores a 220Kbps e 300Kbps, para os dois bit rates, respectivamente. Para pacotes de tamanho superior o valor efectivo começa a estabilizar num valor superior ao nominal, verificando-se taxas de transmissão até 333Kbps e 554Kbps, para pacotes de 4KB e ambos os bit rates, 300 e 500 Kbps, respectivamente.

Curiosamente, parece que estes dois bit rates resultam exclusivamente da diferente forma de comunicação imposta pelo uso dos pacotes elementares de 2 e 7 bytes, que por sua vez, resultam em dois mecanismos de correcção de erros diferentes, tal como indicado no manual do modem DCB500.

4.2. Medição do Round Trip Delay (RTD)

Em telecomunicações, o termo Round Trip Delay (RTD) ou Round Trip Time (RTT) significa o tempo passado para que um sinal ou uma mensagem viaje de um emissor para um receptor e volte de novo, ou seja, o tempo passado desde o instante imediatamente antes de enviar a mensagem e o instante imediatamente depois de a mesma mensagem ser completamente recebida no emissor [35]. Os testes de medição do RTD permitem incluir nos valores medidos o tempo de retransmissão no receptor, isto é, o tempo que medeia entre a recepção da mensagem e o seu reenvio, e permitem ter uma ideia do overhead computacional que o processamento das mensagens acarreta.

Para a medição do RTD foi utilizado o processo apresentado na Fig. 52.

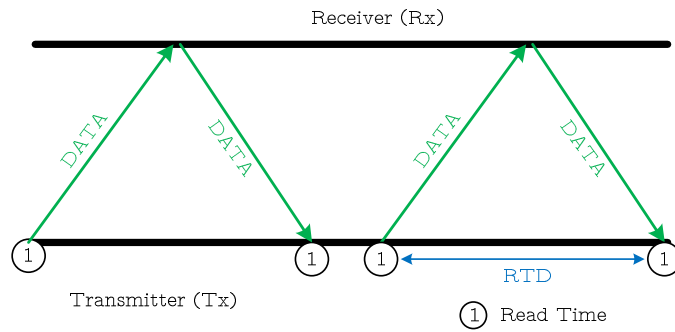


Fig. 52 – Processo de medida do RTD

É capturado o valor do *timer* imediatamente antes de enviar a mensagem e novamente imediatamente depois de ser recebida essa mesma mensagem de volta. De seguida é efectuada a subtracção destes dois valores, obtendo-se assim o tempo de ida e volta da mensagem (RTD).

Foram efectuadas medidas do RTD de mensagens de tamanho compreendido entre 1 e 4096 bytes. Os resultados obtidos são os apresentados na Fig. 53 e Fig. 54. De notar a escala horizontal logarítmica dos gráficos de mensagens de tamanho compreendido entre 64 e 4KB.

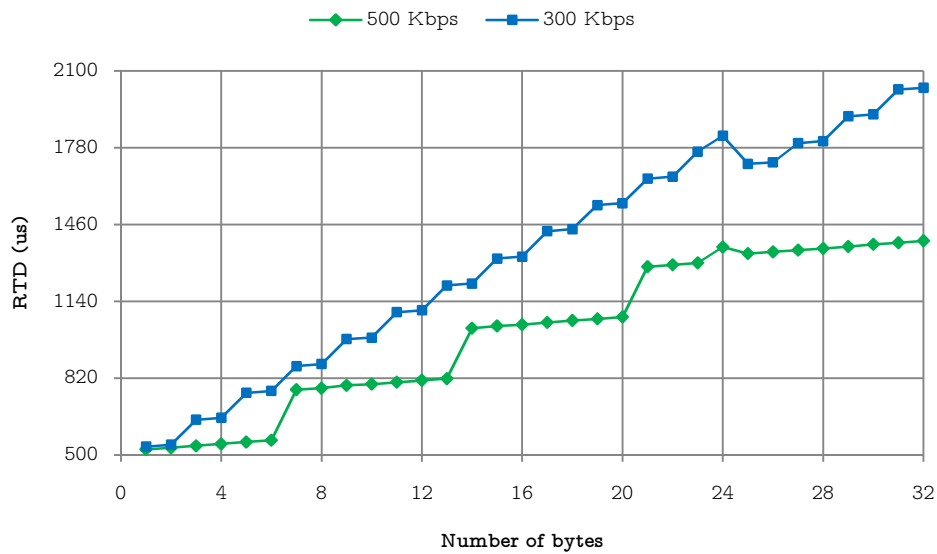


Fig. 53 – RTD medido para mensagens entre 1 e 32 bytes – Stream Mode

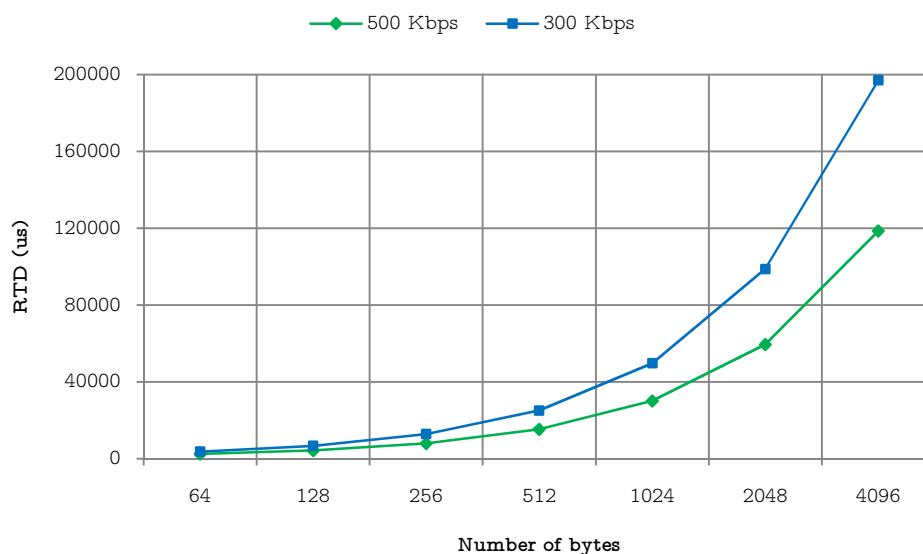


Fig. 54 – RTD medido para mensagens entre 64 e 4096 bytes – Stream Mode

Observando e comparando os resultados medidos verifica-se que os valores de Round Trip Delay são aproximadamente o dobro dos valores medidos para o tempo de transmissão. Este resultado era o esperado, pois os nós são bastante rápidos a retransmitir as mensagens recebidas.

4.3. Comparação das medidas efectuadas, com e sem arbitragem

As experiências apresentadas nas secções anteriores, secções 4.1 e 4.2, foram repetidas, agora com arbitragem activa.

O modem transdutor DCB500 permite a configuração do uso ou não de arbitragem, *Arbitration Enable* e *Arbitration Disable*, respectivamente. Dentro da *Arbitration Enable* permite ainda configurar entre arbitragem aleatória (*Random Arbitration*) e arbitragem com prioridades.

Quando activa o processo de arbitragem não inclui qualquer mecanismo de retransmissão automática, que deve ser efectuada pela aplicação. Os detalhes sobre o processo de arbitragem são, no entanto, desconhecidos.

O fabricante afirma que este segue a técnica CSMA/CA. Contudo, o método utilizado é não persistente pois os nós não esperam que o meio esteja livre para transmitir. Nestas condições não é explícito o que se deve entender pela técnica CA (Collision Avoidance) presente no DCB500. Segundo o manual os nós escutam o barramento por aproximadamente 50µs antes de transmitir, para verificar se este se encontra livre (arbitragem desactivada) ou alternativamente executam o protocolo de arbitragem (arbitragem activa).

Na arbitragem com prioridades, a prioridade de cada mensagem define-se pelos 4 bits menos significativos do primeiro byte da mensagem (excluindo o byte de comando). A mensagem de menor prioridade será a que tiver os 4 bits menos

significativos iguais a “1111b” e a de maior prioridade será a mensagem que tiver estes 4 bits iguais a “0000b” [28].

Os resultados obtidos, tempos de transmissão com e sem arbitragem e RTD com e sem arbitragem são de seguida comparados. De notar a escala horizontal logarítmica dos gráficos de mensagens de tamanho compreendido entre 64 e 4KB.

Nos gráficos seguintes as abreviaturas apresentadas são:

- AD – Arbitration Disable
- AE – Arbitration Enable

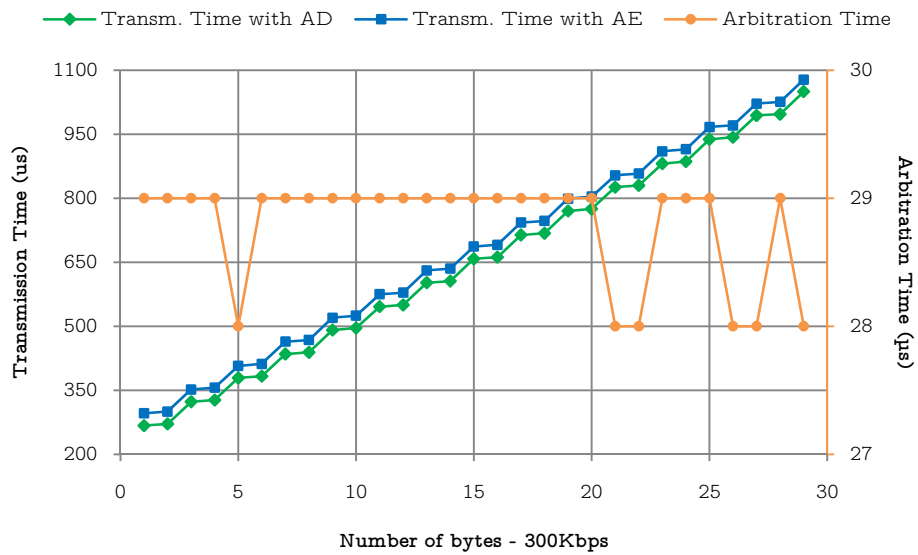


Fig. 55 – Tempo de transmissão – 1 a 29 bytes – Single Message Mode – 300Kbps

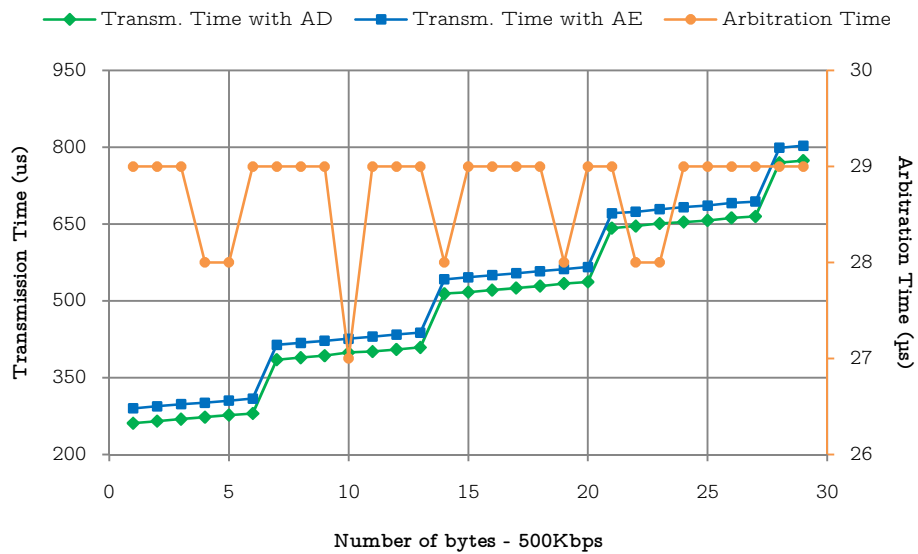


Fig. 56 – Tempo de transmissão – 1 a 29 bytes – Single Message Mode – 500Kbps

Os gráficos apresentados na Fig. 55 e Fig. 56 expõem os resultados obtidos para o modo *Single Message* para 300 e 500Kbps, respectivamente. Verifica-se que todos os pontos medidos com arbitragem activa são superiores aos medidos anteriormente (arbitragem inactiva). Os pontos laranja representam os tempos de arbitragem medidos para cada pacote de dados enviado.

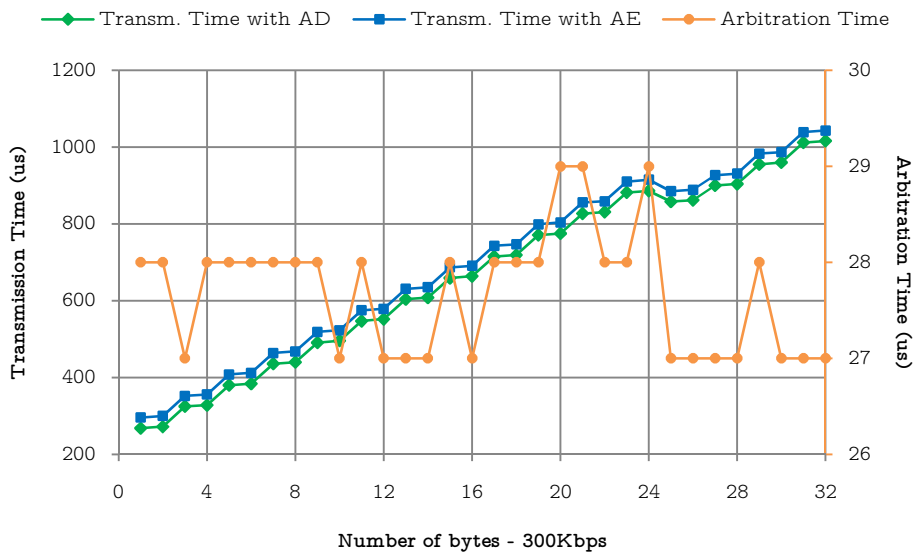


Fig. 57 – Tempo de transmissão – 1 a 32 bytes – Stream Message Mode – 300 Kbps

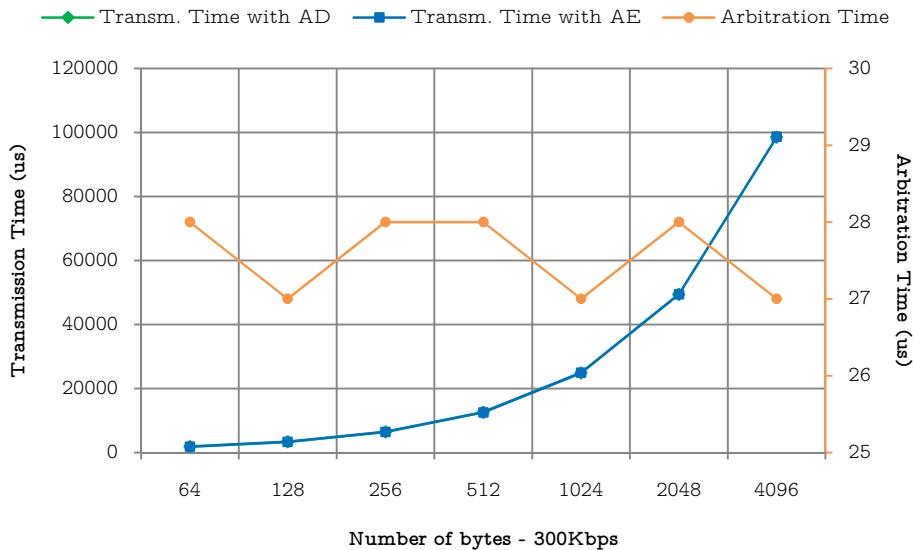


Fig. 58 – Tempo de transmissão - 64 e 4096 bytes – Stream Message Mode – 300Kbps

Nos gráficos da Fig. 57 e Fig. 58 estão representadas as mesmas medidas mas desta vez para o modo *Stream Message*, para a taxa de transmissão de 300Kbps.

Por outro lado, na Fig. 59 e Fig. 60 estão representadas os valores obtidos para o mesmo modo agora para uma taxa de transmissão nominal de 500Kbps.

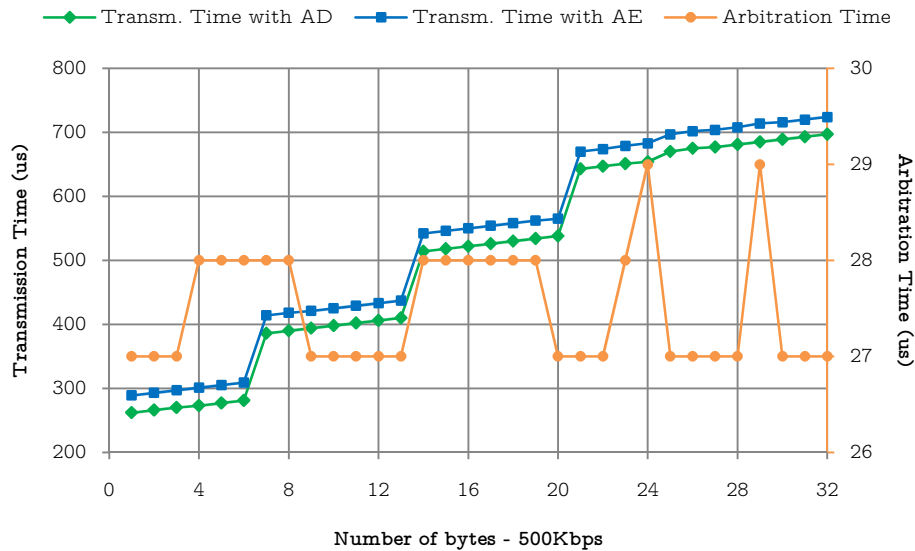


Fig. 59 – Tempo de transmissão – 1 a 32 bytes – Stream Message Mode – 500Kbps

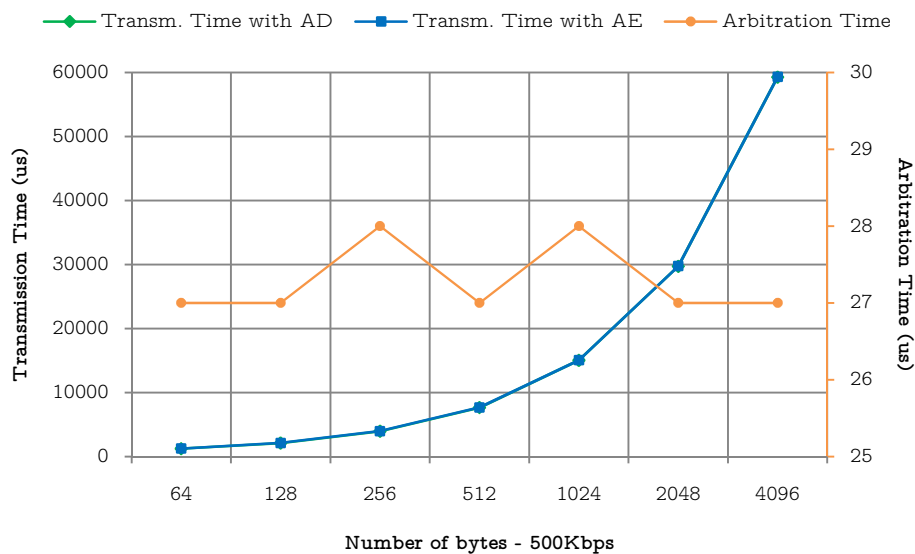


Fig. 60 – Tempo de transmissão – 64 a 4096 bytes – Stream Message Mode – 500Kbps

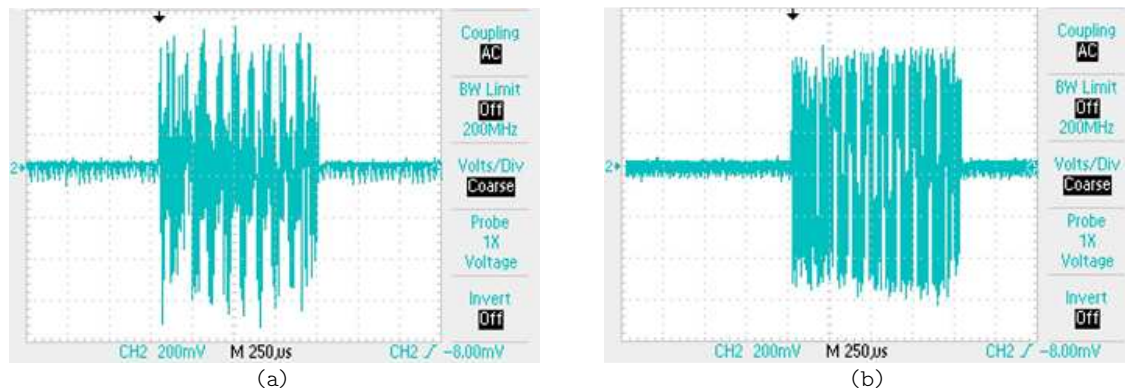
Todos os offsets calculados para o tempo de arbitragem têm resolução de 1 μ s e todos se encontram dentro dessa margem. Assim, pode afirmar-se que estes valores não variam com o número de bytes enviados, sendo assim independente do tamanho do pacote de dados, com os seguintes valores.

Tabela 9 – Offsets determinados – Single e Stream Mode – 300 e 500Kbps

Bit Rate	Single Message		Stream Message	
	Média	Desvio Padrão	Média	Desvio Padrão
300Kbps	28,8	0,4	27,7	0,6
500Kbps	28,7	0,5	27,5	0,6

Em todos os casos, verificou-se um offset de 29 μ s para *Single Message* e 28 μ s para *Stream Message* ($\pm 1\mu$ s). Considerando que os valores medidos sem arbitragem incluem 50 μ s de tempo de escuta do barramento e os valores medidos com arbitragem activa têm um offset adicional de aproximadamente 30 μ s, pode concluir-se que o processo de arbitragem leva cerca de 80 μ s, o que é próximo do valor presente no manual técnico do DCB500, que indica um tempo do processo de arbitragem de aproximadamente 100 μ s. No entanto, como as experiências foram efectuadas apenas com 2 nós não foi possível concluir se este tempo varia com o número de nós ligados ao barramento.

O offset calculado foi ainda confirmado pela obtenção de algumas imagens da actividade no barramento Powerline (Fig. 61).

**Fig. 61** – Transmissão de 31 bytes. (a) – Arbitration Disable. (b) – Priority Arbitration

Tal como já afirmado, verifica-se que o tempo de transmissão é maior com arbitragem activa e verifica-se que o tempo adicional dispendido com arbitragem activa é utilizado em comunicações que será usada pelos nós para chegar a um consenso sobre quem tem a prioridade mais alta.

Adicionalmente é possível calcular o Bit Rate interno, ou seja, a velocidade a que os pacotes de dados são transmitidos internamente. Medindo o incremento temporal provocado pela adição de apenas um byte de dados na transmissão sem que seja incrementado o número de pacotes elementares, isto é, numa das zonas planas do gráfico dos tempos de transmissão.

$$\text{Bit Rate interno} = \frac{8 \text{ bit}}{\Delta T} [\text{Kbps}] \quad (9)$$

Assim, obtém-se um bit rate interno de 2000Kbps, ou seja, 2Mbps, para ambos os bit rates.

4.4. Colisões

Neste lote de testes pretende-se averiguar o comportamento do canal de comunicação Powerline face a colisões de mensagens, contemplando três situações distintas – Transmissão em simultâneo e Transmissão desfasada, com tempo de desfasamento comparável ao de arbitragem (aproximadamente 50us) e com um desfasamento bastante menor (aproximadamente 15us) – e os três casos possíveis de configuração da arbitragem – Arbitration Disable, Random Arbitration e Priority Arbitration. Provocaram-se colisões entre as mensagens de cada nó e mediu-se a quantidade de mensagens perdidas por cada um deles.

O modem transdutor DCB500 permite a activação da interrupção de colisão, ou seja, cada vez que detecta uma colisão no barramento sinaliza o Host com uma interrupção e o Host lê o registo de interrupção do modem. Se o código de interrupção lido for o correspondente à interrupção de colisão é incrementado o contador de mensagens perdidas. A Fig. 62 ilustra esta situação.

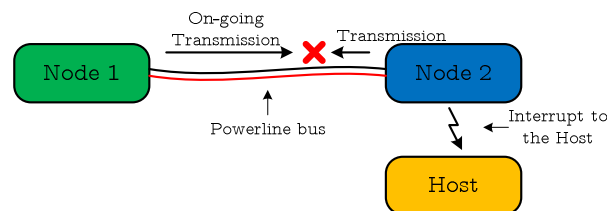


Fig. 62 – Ilustração de uma colisão e sinalização ao Host

4.4.1. Transmissões em simultâneo

Com o objectivo de estudar o comportamento do barramento Powerline, o primeiro teste consistiu no envio de duas mensagens tanto em simultâneo quanto possível ($\pm 1\mu s$), usando para tal um par de sinais de sincronização externos e registando a quantidade de mensagens perdidas por cada nó.

Foi usado apenas o modo *Stream Message* dada a possibilidade de efectuar uma variação mais ampla do tamanho do pacote de dados enviado. São enviadas 2000 mensagens e calculada a percentagem de mensagens perdidas por colisão ou perda do acesso ao barramento devido ao processo de arbitragem.

Para a medição destes valores foi desenvolvido o seguinte algoritmo, apresentado na Fig. 63.

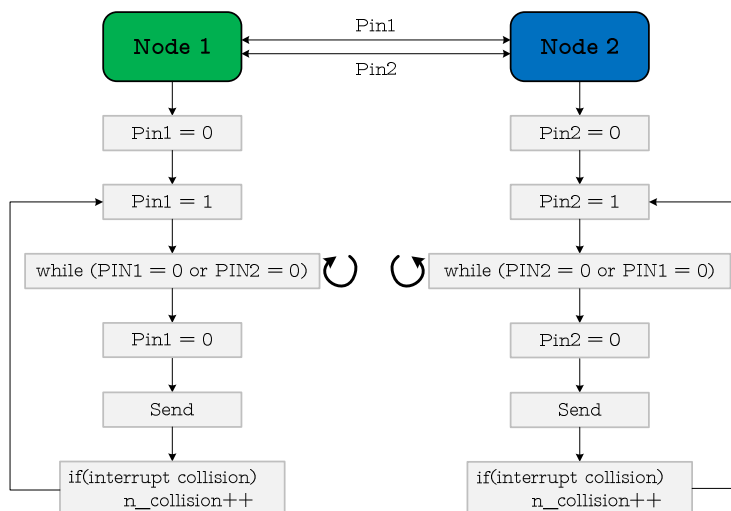


Fig. 63 – Algoritmo para a medição de mensagens perdidas – Transmissão simultânea

Nos gráficos seguintes são apresentados os resultados obtidos nestas medições, abordando os três casos de arbitragem apresentados.

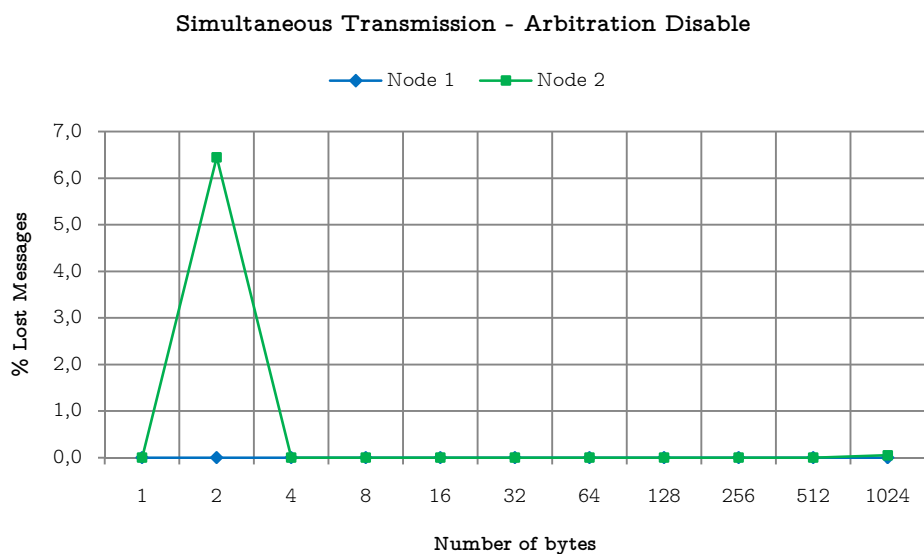


Fig. 64 – Duas transmissões simultâneas – Arbitration Disable

No primeiro caso, em que não existe arbitragem no acesso ao barramento (Fig. 64), a percentagem de colisões detectadas é bastante baixa, para ambos os nós. No entanto, as mensagens parecem também não ser entregues ao outro nó. Este aspecto necessita de ser revisitado posteriormente com um canal de comunicação com um mínimo de três nós para que se possa averiguar a recepção correcta ou não das mensagens.

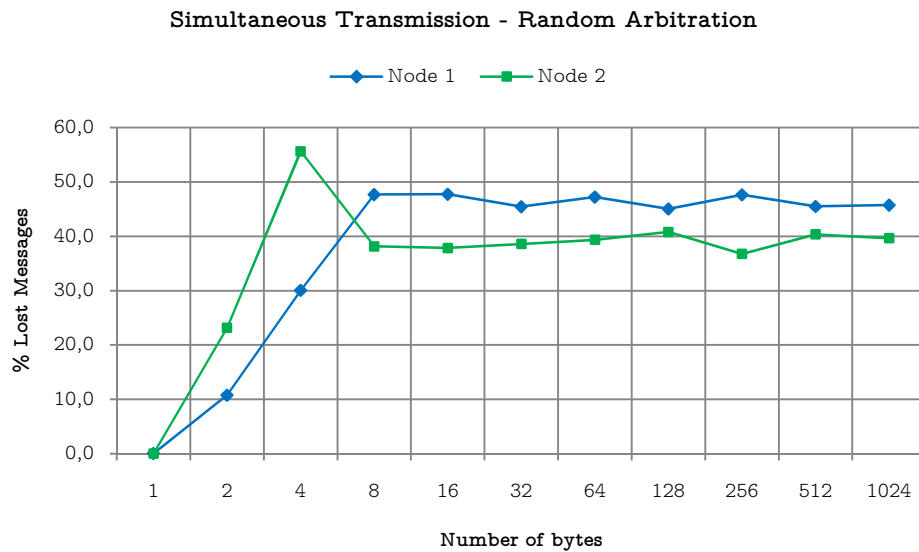


Fig. 65 – Duas transmissões simultâneas – Random Arbitration

Para a arbitragem aleatória (Fig. 65), verifica-se o esperado, ou seja, que existem situações de perda de pacotes por parte de ambos os nós. Para pacotes de dados de dimensão inferior a 8 bytes verificam-se mais perdas por parte do nó 2, ao passo que para mensagens maiores as perdas medidas são sempre maiores para o nó 1. No entanto, o modo de funcionamento da aleatoriedade da arbitragem é também desconhecido.

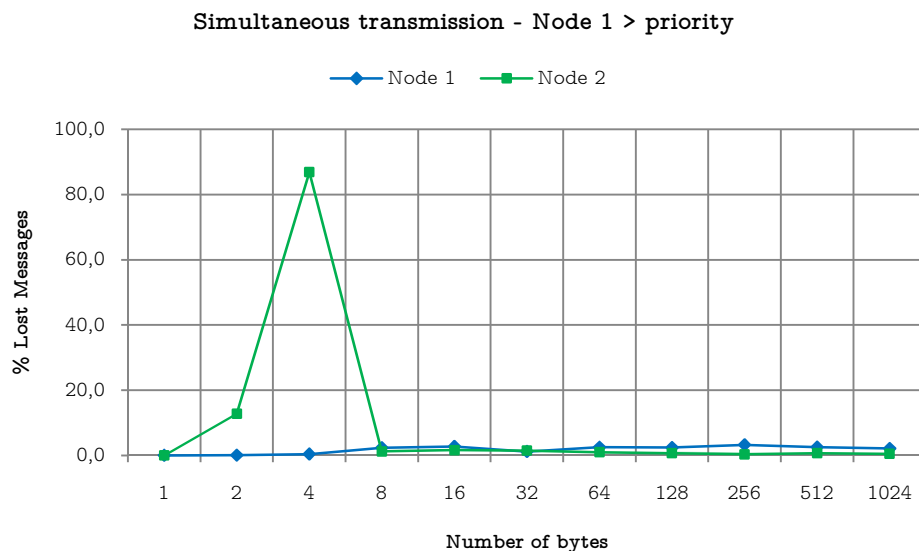


Fig. 66 – Duas transmissões simultâneas – Node 1 > priority

Para mensagens de tamanho igual ou superior a 8 bytes, assim como para 1 byte, verificou-se que praticamente não são detectadas colisões para ambos os nós, com algumas situações em que o nó mais prioritário chega a detectar um pouco mais do que o outro de menor (inversão de prioridades). Esta situação não era espectável

visto que o nó mais prioritário deveria sempre ganhar o processo de arbitragem enquanto o nó menos prioritário deveria sempre detectar a colisão e perder a arbitragem. Este comportamento manteve-se trocando as prioridades das mensagens enviadas pelos nós, assim como trocando os interfaces DCB500, indiciando uma arbitragem deficiente.

4.4.2. Desfasamento maior que o offset de arbitragem

Neste caso o objectivo é verificar quais os efeitos das colisões, em termos de percentagem de mensagens perdidas, quando se enviam mensagens para o barramento desfasadamente, com o desfasamento maior que a duração do offset medido para o processo de arbitragem.

Sendo o offset determinado na secção 4.3 de cerca de 28 μ s, usou-se um desfasamento de 50 μ s no envio das mensagens. Assim, o nó 1 envia a sua mensagem 50 μ s após o nó 2. O algoritmo desenvolvido para esta experiência é apresentado na Fig. 67.

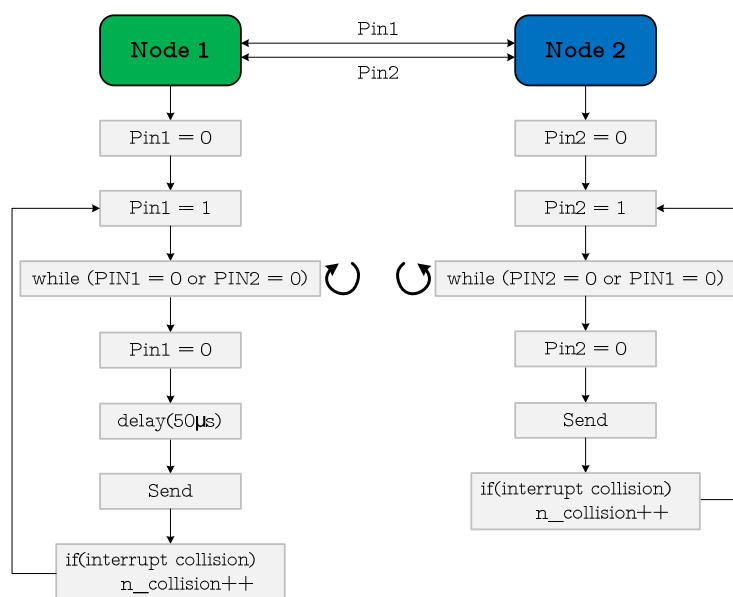


Fig. 67 – Algoritmo desenvolvido – Desfasamento maior que o offset medido

De seguida são apresentados os resultados obtidos nesta experiência.

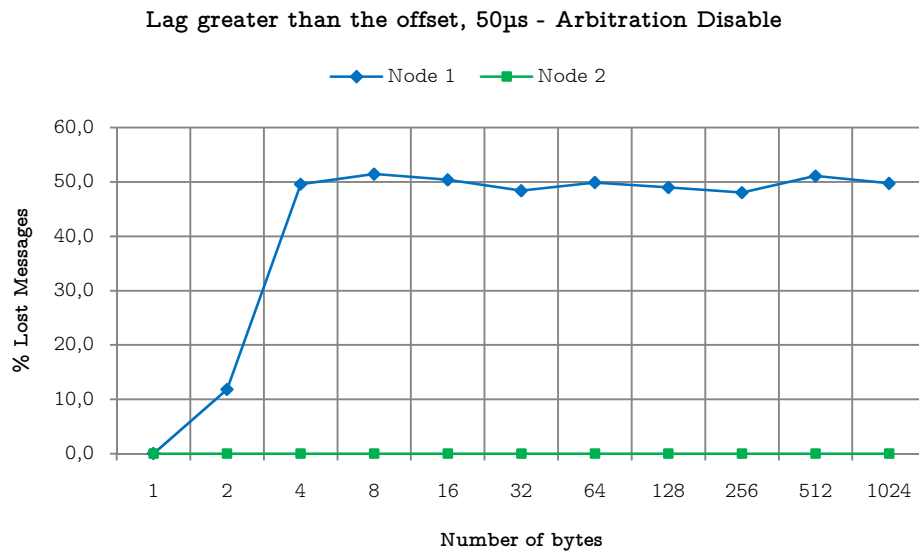


Fig. 68 – Desfasamento maior que o offset, 50 μ s – Arbitration Disable

Na situação do envio das mensagens do nó 1 com desfasamento maior que o offset medido espera-se que este tenha uma percentagem de perda de mensagens superior ao nó 2, visto que este último já se encontra a terminar o processo de “escuta” do barramento quando chega a mensagem do nó 1 ao barramento.

Analisando o gráfico da situação em que não existe arbitragem e o desfasamento das mensagens do nó 1 é superior ao tempo de arbitragem (Fig. 68) verifica-se o esperado, que o nó 1 perde sempre mais mensagens que o nó 2. Sendo que, para pacotes de dimensão superior ou igual a 4 bytes observa-se uma perda de aproximadamente 50% das mensagens do nó 1 e não são registadas perdas por parte do nó 2, para qualquer tamanho de mensagem de dados.

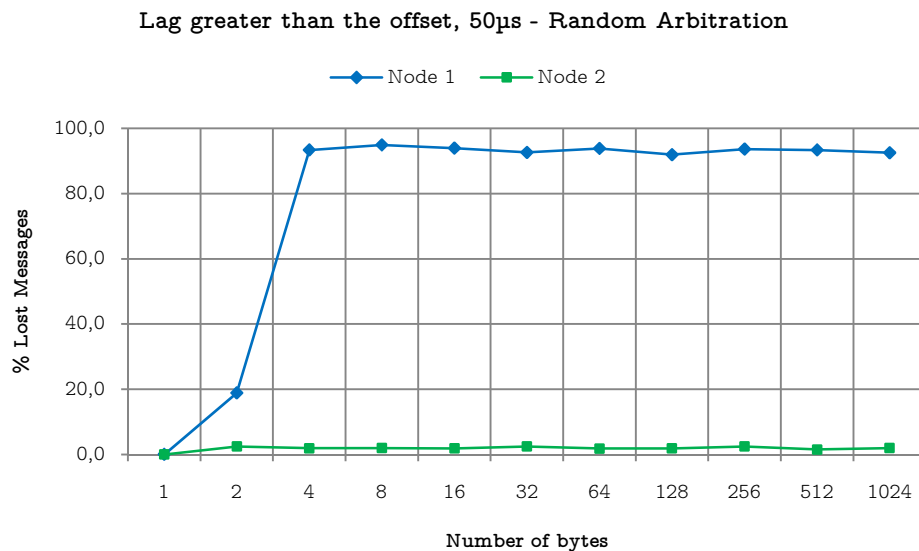


Fig. 69 – Desfasamento maior que o offset, 50 μ s – Random Arbitration

No caso de arbitragem aleatória (Fig. 69), o nó 1, como chega atrasado, perde quase a totalidade das arbitragens, enquanto o nó 2 apenas perde algumas (poucas) vezes. No caso da arbitragem com prioridade (Fig. 70), o nó 1 perde a totalidade das arbitragens para mensagens com tamanho igual ou superior a 4 bytes, enquanto o nó 2 não apresenta perda. Embora este resultado seja contrário às prioridades estabelecidas, é explicado pelo atraso com que o nó chega ao barramento, já com a comunicação do nó 2 em curso. Este comportamento também se manteve trocando prioridades e interfaces físicos.

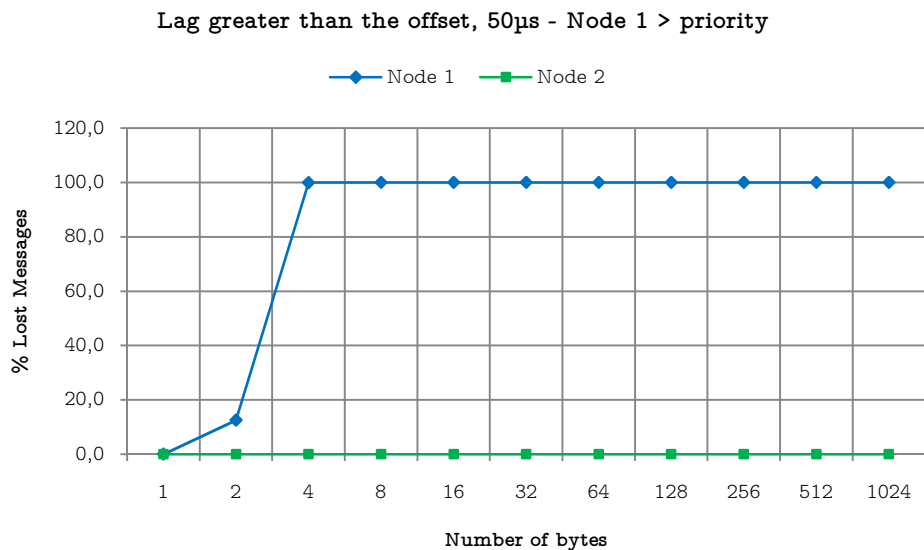


Fig. 70 – Desfasamento maior que o offset, 50 μ s – Node 1 > priority

4.4.3. Desfasamento menor que o offset de arbitragem

Este teste consistiu em repetir o teste anterior mas com um desfasamento menor que a duração do offset de arbitragem. Assim, forçou-se o nó 1 a transmitir durante o processo de “escuta” do barramento, ou seja, 15 μ s depois do nó 2.

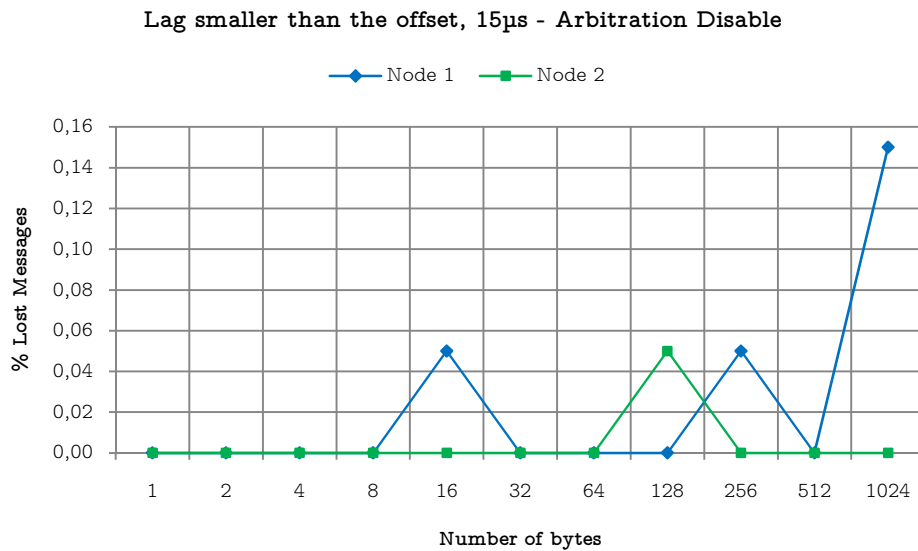


Fig. 71 – Desfasamento menor que o offset, 15 μ s – Arbitration Disable

Neste caso, em que o desfasamento do nó 1 é menor que o tempo de “escuta” do barramento verifica-se que as perdas detectadas são bastante pequenas. No entanto, tal como no caso de *Arbitration Disable* e envio em simultâneo, as mensagens parecem também não ser entregues ao outro nó. Será assim também visitar posteriormente este caso com um canal de comunicação com um mínimo de três nós para que se possa averiguar a recepção correcta ou não das mensagens.

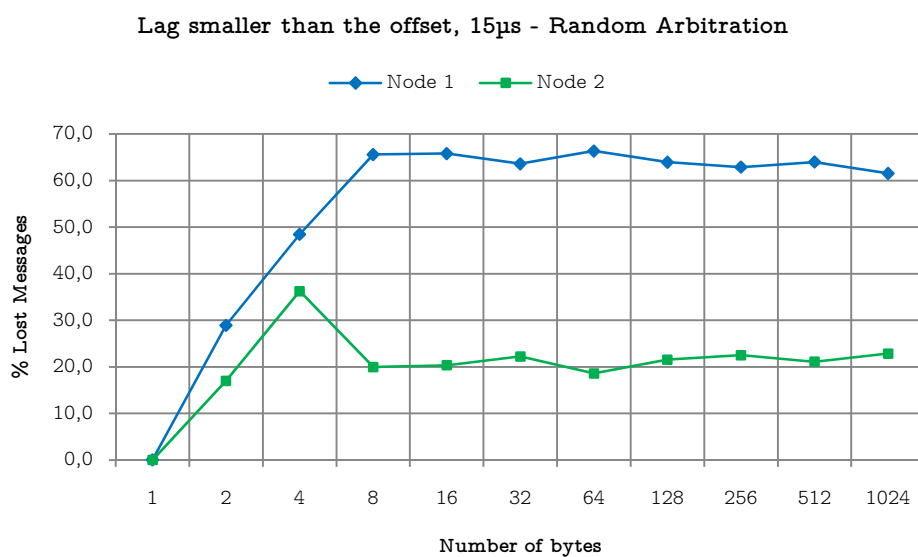


Fig. 72 – Desfasamento menor que o offset, 15 μ s – Random Arbitration

No caso em que o sistema é programado para que o nó 1 transmita 15 μ s após o nó 2 e a arbitragem seja aleatória (Fig. 72), verificam-se perdas de mensagens por parte de ambos os nós, embora o nó 1 perca sempre mais mensagens do que o nó 2. Este comportamento é justificável pela chegada do nó 1 ao barramento já quando o nó 2 se encontra no processo de escuta do barramento.

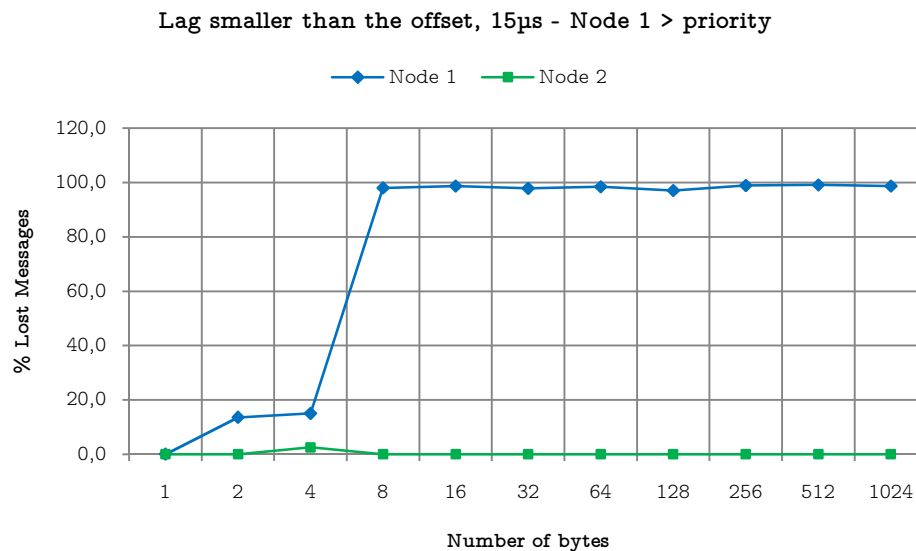


Fig. 73 – Desfasamento menor que o offset, 15 μ s – Node 1 > priority

Quando o desfasamento do nó 1 é igual a 15 μ s e os nós se encontram programados com prioridades (Fig. 73), sendo que o nó 1 é o mais prioritário, verifica-se que o processo de arbitragem não funciona e que, pelo facto do nó 1 iniciar a sua tentativa de transmissão depois de o nó 2 já se encontrar no processo de “escuta” do barramento, o nó 1 perde menos de 20% das suas mensagens, para os casos em que os pacotes de dados são iguais ou inferiores a 4 bytes, e perde aproximadamente 100% das suas mensagens quando o pacote de dados que tenta enviar é igual ou superior a 8 bytes.

É de salientar ainda que todos estes testes foram repetidos trocando prioridades e desfasamentos entre nós e todos os comportamentos apresentados se mantiveram. Repetiram-se ainda as medidas efectuando a troca física dos modems transdutores DCB500 e verificou-se que os comportamentos foram igualmente mantidos e assim confirmados.

5. CONCLUSÕES E TRABALHO FUTURO

5.1. Conclusões e comentários finais

O uso da tecnologia de comunicação Powerline (PLC) em telemetria, domótica e redes domésticas é já uma realidade comum. Contudo, acredita-se que esta tecnologia é também aplicável e benéfica para o domínio automóvel, contribuindo fortemente para a redução de cablagem e consequentemente do peso de novos modelos de veículos. Este benefício é bastante bem compreendido para os veículos normais, que possuem motor de combustão, mas é ainda de mais fácil compreensão quando empregue em veículos eléctricos [36].

Hoje em dia a comunicação automóvel é dominada pelas soluções já bem conhecidas e robustas, tais como, LIN, CAN e FlexRay. Contudo, uma solução baseada em PLC deve ser comparada com estas soluções em aspectos essenciais como fiabilidade, largura de banda e tempos de resposta.

Recentemente surgiram algumas soluções prontas a usar (COTS - Commercial Off The Shelf) para comunicação Powerline sobre linhas DC mas ainda assim, necessitam de uma avaliação cuidada e individual não apenas em termos de problemas relacionados com interferências electromagnéticas (EMI) e robustez ao ruído no barramento DC mas também em termos de comportamento temporal e largura de banda [36].

O trabalho apresentado nesta dissertação é um estudo experimental preliminar de avaliação do comportamento temporal e de acesso ao meio dos adaptadores DCB500, destinados a barramentos DC, construídos pela empresa Yamar Electronics.

Quanto ao comportamento temporal, os resultados mostram a introdução de um considerável *overhead* na comunicação em modo *Single Message*, que está limitado ao envio máximo de 31 bytes de dados. No entanto, este *overhead* começa a desvanecer-se quando é usado o modo *Stream Message* que suporta transferências de maiores quantidades de dados. Verificou-se que as velocidades de transmissão disponibilizadas pela DCB500 são fruto da forma como os dados são enviados, ou seja, devido ao envio dos dados em pacotes elementares e do mecanismo de correcção de erros utilizado. Foi ainda possível determinar a velocidade real a que os pacotes são transmitidos no barramento DC, que se verificou ser 2Mbps.

O processo de arbitragem para acesso ao barramento, um aspecto importante no comportamento de tempo-real de um sistema de comunicação, geralmente funciona correctamente e como esperado mas, em alguns casos, apresenta alguns comportamentos curiosos, o que parece indicar um pobre suporte de comunicação com prioridades [36]. No entanto, é necessário levar a cabo mais testes para uma completa avaliação destes comportamentos, em particular, aspectos como o suporte de acesso assíncrono com arbitragem baseada em prioridades, à semelhança de CAN, o que seria muito desejável para aplicação em sistema de automóveis com requisitos de tempo-real, tal como airbag, ESP, etc.

5.2. Trabalho futuro

Com uma análise mais profunda dos resultados obtidos surgiu a necessidade de efectuar novos testes com o objectivo de confirmar e/ou averiguar a razão de alguns comportamentos verificados que, ou não vão ao encontro do esperado ou podem ser

melhor suportados com a introdução de um terceiro nó no canal de comunicação já existente de forma a permitir a verificação das mensagens válidas no barramento, ou seja, permitindo assim verificar se as mensagens são efectivamente entregues ao outro nó ou se por outro lado existem colisões não detectadas, o que parece acontecer em alguns casos.

De seguida, e analisando todos os resultados obtidos e caracterizando o processo de arbitragem presente nos adaptadores DCB500, seria interessante desenvolver um protocolo de nível superior com uma política de retransmissão automática de mensagens. No entanto é necessário encontrar uma forma de detectar o meio livre, visto que o manual não é claro acerca deste aspecto e depois de questionar a Yamar foi facultada a informação que não existe nenhum pino exterior ou sinal que disponibilize esta informação. Assim, será necessário efectuar um estudo mais aprofundado do hardware DCB500 para que se averigüe se em algum ponto da board se pode encontrar algum sinal que permita obter esta informação.

Ao nível de interferências podem ser repetidas as medidas já efectuadas usando fontes de interferência e um atenuador, podendo ser utilizado o atenuador construído também pela Yamar Electronics, com o objectivo de permitir estes mesmos testes. Como fontes de interferência podem ser utilizadas, por exemplo, telemóveis em comunicação junto do barramento de comunicação e ainda uma topologia de alimentação em árvore, com ramos de vários comprimentos (pelo menos 5-10 ramos) ligados ao barramento e de seguida efectuar os mesmos testes com cargas comutadas (ligar e desligar lâmpadas, motores, etc.).

Futuramente seria também interessante estudar a possibilidade de substituir o fio de alimentação por um par entrançado de modo que o sinal transmitido seja diferencial, tornando a comunicação bastante mais imune a ruído.

6. REFERÊNCIAS

- [1] T. Nolte, H. Hansson, and L. L. Bello, "Automotive Communications - Past, Current and Future," in *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on.* vol. 1 Catania, 2005, pp. 985 - 992.
- [2] Wikipedia. "Local Interconnect Network." Internet: http://en.wikipedia.org/wiki/Local_Interconnect_Network, [May 18, 2009]
- [3] Wikipedia. "FlexRay." Internet: <http://en.wikipedia.org/wiki/FlexRay>, [May 18, 2009]
- [4] Wikipedia. "Media Oriented Systems Transport." Internet: http://en.wikipedia.org/wiki/Media_Oriented_Systems_Transport, [May 18, 2009]
- [5] Wikipedia. "Power line communication." Internet: http://en.wikipedia.org/wiki/Power_line_communication, [May 19, 2009]
- [6] L. F. Montoya, "Power Line Communications Performance: Overview of the Physical Layer of Available protocols," Florida: University of Florida, 1998.
- [7] "X-10 Communications Protocol and Power Line Interface PSC04 & PSC05, Revision 2.4," X-10 PRO.
- [8] M. Galeev. "Embedded.com - Home networking with Zigbee." Internet: http://www.embedded.com/columns/technicalinsights/18902431?_requestid=364220, [May 20, 2009]
- [9] M. E. Hazen, "The Technology Behind HomePlug AV Powerline Communications," in *Computer.* vol. 41 Los Alamitos, CA, USA: IEEE Computer Society Press, 2008, pp. 90-92.
- [10] S. Gardner, B. Markwalter, and L. Yonge, "HomePlug Standard Brings Networking to the Home," in *Communication Systems Design.* vol. 6, 2000.
- [11] "Welcome to the world of dLAN®." Internet: <http://www.devolo.com/>, [May 12, 2009]
- [12] F. Benzi, T. Facchinetti, T. Nolte, and L. Almeida, "Towards the Powerline Alternative in Automotive Applications," in *7th IEEE International Workshop on Factory Communication Systems (WFCS - Work-in-progress session)* Dresden, Germany, 2008, pp. 259-262.
- [13] C. i. A. (CiA). "CAN in cars using a single DC power-line for communication." Internet: <http://www.can-cia.org/index.php?id=888&L=http>, [May 22, 2009]
- [14] Bosch, "CAN Specification, Version 2.0B," 1991.
- [15] Microchip Technology Inc, "Controller Area Network (CAN) Basics," DS00713A, 1999.

- [16] Wikipedia. "Controller-area network." Internet: http://pt.wikipedia.org/wiki/Controller_Area_Network, [May 15, 2009]
- [17] "The CAN Protocol Tour." Internet: <http://www.kvaser.com/can/protocol/index.htm>, [Feb. 05, 2009]
- [18] LIN Consortium, "LIN Specification Package, Revision 2.0," Munich, Germany 2003.
- [19] "Introduction to the LIN bus." Internet: <http://www.kvaser.com/can/misc/lin/lin-bus.htm>, [May 23, 2009]
- [20] Clemson Vehicular Electronics Laboratory. "AUTOMOTIVE BUSES." Internet: http://www.cvel.clemson.edu/auto/auto_buses01.html, [May 24, 2009]
- [21] F. Luo, Z. Chen, J. Chen, and Z. Sun, "Research on FlexRay Communication System," in *IEEE Vehicle Power and Propulsion Conference (VPPC)* Harbin, China, 2008, pp. 1-5.
- [22] X. Yi-Nan, I. G. Jang, Y. E. Kim, J. G. Chung, and L. Sung-Chul, "Implementation of FlexRay protocol with an automotive application," in *SoC Design Conference, 2008. ISOC '08. International*, 2008, pp. II-25-II-28.
- [23] Wikipédia. "FlexRay." Internet: <http://pt.wikipedia.org/wiki/FlexRay>, [May 24, 2009]
- [24] "FlexRay Introduction." Internet: http://www.tzm.de/en/FlexRay/FlexRay_Introduction.html, [May 24, 2009]
- [25] Elektrobit. "FlexRay: The Protocol with the Frame." Internet: http://www.elektrobit.com/what_we_deliver/automotive_software/technologies_and_standards/flexray/the_protocol_with_the_frame, [May 25, 2009]
- [26] N. Instruments. "FlexRay Automotive Communication Bus." Internet: <http://zone.ni.com/devzone/cda/tut/p/id/3352>, [May 25, 2009]
- [27] Yamar Electronics. "Profile of Yamar electronics DC-BUS battery power line communication." Internet: <http://www.yamar.com/profile.html>, [May 25, 2009]
- [28] Yamar Electronics, "DCB500A Evaluation Board Manual," 2006.
- [29] Yamar Electronics. "SIG40 - UART Over Power line Transceiver." Internet: <http://www.yamar.com/SIG40.html>, [May 25, 2009]
- [30] Yamar Electronics, "SIG40 - UART Communication Over Battery Power Lines," 2008.
- [31] Yamar Electronics, "DCAN250 - CAN over Battery Power Line Communication," 2007.

- [32] Yamar Electronics. "DCAN250 Transceiver for CAN-bus Over PLC Network." Internet: <http://www.yamar.com/DCAN250.html>, [May 25, 2009]
- [33] Yamar Electronics. "DCB500 UART / SPI Serial PLC Modem Transceiver." Internet: <http://www.yamar.com/DCB500.html>, [May 25, 2009]
- [34] Yamar Electronics, "DCB500 - Transceiver for Powerline Communication," 2008.
- [35] "What is round-trip time?" Internet: http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci1250602,00.html, [May 30, 2009]
- [36] P. Silva, L. Almeida, D. Caprini, T. Facchinetti, F. Benzi, and T. Nolte, "A Study of the Real-Time Behavior of DC-Powerline for Automotive Systems," in *ETFA 2009 - 14th IEEE International Conference on Emerging Technologies and Factory Automation*, Mallorca, Spain, 2009 (em submissão).